9-2004

Every Polynomial-Time 1-Degree Collapses If And Only If P=PSPACE

Stephen A. Fenner
University of South Carolina - Columbia, fenner@cse.sc.edu

Stuart A. Kurtz

James S. Royer

Publication Info

Published in *The Journal of Symbolic Logic*, Volume 69, Issue 3, 2004, pages 713-741.
http://www.aslonline.org/journals-journal.html

THE JOURNAL OF SYMBOLIC LOGIC
Volume 69, Number 3, Sept. 2004

EVERY POLYNOMIAL-TIME 1-DEGREE COLLAPSES IF AND ONLY IF P = PSPACE

STEPHEN A. FENNER, STUART A. KURTZ, AND JAMES S. ROYER

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

- *1-equivalent* if and only if each is m-reducible to the other by one-one reductions;
- *p-invertible equivalent* if and only if each is m-reducible to the other by one-one, polynomial-time invertible reductions; and
- *p-isomorphic* if and only if 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. Introduction. In concrete applications of polynomial-time reductions (e.g., in NP-completeness proofs [GJ79]) m-reducibility[1] is by far the most common reducibility notion employed. These specific m-reductions tend to have strong properties: they are almost always honest[2] usually length-increasing, and frequently one-one. The usual interpretation of one set, A, being m-reducible to another, B, is that A is computationally no more difficult than B since from any decision procedure for B we can construct a decision procedure for A of polynomially related complexity. But this interpretation is also supported by polynomial-time Turing reducibility, a much weaker reducibility. The m-reducibility of A to B thus suggests a stronger relation between A and B than implied by the conventional interpretation, and indeed there are cases where we are able to obtain additional useful information from the strength of these reducibilities. For example, it is known that the m-complete sets for deterministic exponential-time are pairwise one-one, length-increasing equivalent [Ber77].

Received November 23, 2000; revised February 29, 2004.
Research for the first author is supported in part by NSF grant CCR-9209833.
Research for the third author is supported in part by NSF grants CCR-89011154 and CCR-9522987.

[1] Since polynomial-time reducibilities are the focus of this paper, we shall usually omit the "polynomial-time" qualifier when referring to one of these reducibilities and add a "recursive" qualifier when referring to a standard reducibility from general computability theory. For example, m-reducibility means polynomial-time m-reducibility whereas recursive m-reducibility is the usual notion from computability theory.

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

0022-4812/04/6903-0006/$3.90

Berman and Hartmanis [BH77] conjectured that the m-complete sets for NP are pairwise p-isomorphic, that is, that the complete m-degree 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), until the late 1980s 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] proof that every 1-li-degree collapses if and only if (as seems unlikely) P = UP. In this paper, we show that the statements:

(a) Every 1-degree collapses.

(b) Every p-invertible degree collapses.

(c) P = PSPACE.

are all equivalent. In retrospect, the most remarkable aspect of our results is the equivalence of (b) and (c) which we still find counterintuitive.

Some conventions and terminology. For the most part we use standard notation and terminology from computability and complexity theory. Here we introduce a few conventions and some less standard notions.

We identify each element of ω, the natural numbers, with its dyadic representation over $\{\mathbf{0},\mathbf{1}\}$. So, there is a one-to-one correspondence between ω and $\{\mathbf{0},\mathbf{1}\}^*$. We shall freely pun between an element of ω being a number and a string over $\{\mathbf{0},\mathbf{1}\}$. For each $x \in \omega$, $|x|$ denotes the length of x's dyadic representation.

We say that A belongs to the class EXP if and only if there is a polynomial p and deterministic Turing-machine such that the machine decides A and runs within $2^{p(n)}$- time. We say that A belongs to the class UP if and only if there is a polynomial p and polynomial-time decidable predicate $Q(\cdot,\cdot)$ such that $A = \{\, x : (\exists y : |y| \leq p(|x|))\, Q(x,y) \,\}$ and, for each x, there is at most one y such that $Q(x,y)$. UP is clearly a subclass of NP. A function f is *one-way* if and only if f is 1-1, honest, and polynomial-time computable, yet not p-invertible. Independently, Berman [Ber77], Grollmann and Selman [GS84] [GS88] and Ko [Ko85] observed that one-way functions exist if and only if P $\neq$ UP.

Suppose A and B are subsets of ω. When A is m-reducible to B, we write $A \leq_{\mathrm{m}}^{\mathrm{p}} B$, and when A is 1-reducible to B, we write $A \leq_{1}^{\mathrm{p}} B$. We say that A is *length-increasing reducible* to B if and only if there is an f that witnesses $A \leq_{\mathrm{m}}^{\mathrm{p}} B$ and either (i) $|f(x)| > |x|$ for all x, or else (ii) $f = \mathrm{id}_\omega$. We say that A is *1-li-reducible* to B (written: $A \leq_{\text{1-li}}^{\mathrm{p}} B$) if and only if A is 1-1, length-increasing reducible to B. We say that A is *2-tt complete* for a class when A is (polynomial-time) btt-complete for the class and this is witnessed by btt-reductions that employ two-variable tt-conditions exclusively; *1-tt completeness* for a class is defined analogously. We note that if A 1-tt complete for EXP, then it turns out that A is also 1-li-complete for EXP [HKR93].

A function f is said to be *strictly t-space computable* if and only if f is computable by a deterministic Turing machine that runs within a space bound of $t(n)$ on the work tapes *and* the input and output tapes. Strict $\mathscr{O}(t(n))$-space, linear-space, and polynomial-space computability are defined in the obvious way.

Related work. Myhill [Myh55] showed that recursive 1-equivalence is much tighter than one might initially expect: if two sets are so similar that they are recursively 1-equivalent, then they are recursively identical. Formally, the result is:

MYHILL'S THEOREM. *Every two recursively 1-equivalent sets are recursively isomorphic.*

There are a number of complexity theoretic versions of Myhill's Theorem. Dowd [Dow82] has perhaps the strongest of these.

DOWD'S THEOREM. *Every two strictly linear-space 1-equivalent sets are strictly linear-space isomorphic.*

In the theory of polynomial-time reducibilities the closest known analogue to Myhill's Theorem is due to Berman and Hartmanis [BH77].

BERMAN AND HARTMANIS'S THEOREM. *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 strong, perhaps unnecessarily strong. An obvious question is whether either of these hypotheses can be weakened. Ko, Long, and Du [KLD87] showed that under the hypothesis that $P \neq UP$, the p-invertibility hypothesis is indeed necessary.

KO, LONG, AND DU'S THEOREM. *If* $P \neq UP$, *then there are 1-li equivalent sets that fail to be p-isomorphic*[3].

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. By a simple argument, Ko, Long, and Du established

KO, LONG, AND DU'S LEMMA. *If* $P = UP$, *then every two 1-li equivalent sets are p-isomorphic.*

The theorem and lemma thus yield the striking characterization:

COROLLARY. $P = UP$ *if and only if every two 1-li equivalent sets are p-isomorphic.*

The corollary provides a complexity characterization of a degree-theoretic property and thus essentially settles the question whether every 1-li degree collapses.

Our results. We establish analogues of Ko, Long, and Du's Theorem and their Lemma for 1-reductions and p-invertible reductions. We first consider our analogues of their theorem. We show

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

THEOREM 2. *If* $P \neq PSPACE$, *then there are p-invertible equivalent sets that fail to be p-isomorphic*[4].

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 2 indicates that under the assumption that $P \neq PSPACE$, the length-increasing hypothesis of Berman and Hartmanis's theorem is close to tight. (Theorem 2 does not preclude the possibility that "length-nondecreasing" can replace "length-increasing" in the hypothesis of Berman and Hartmanis's theorem. We conjecture that under

[3]Moreover, there are such sets that are 2-tt complete for EXP.

[4]For Theorems 1 and 2 the witnessing sets constructed can to be 2-tt complete for EXP.

a stronger condition than P $\neq$ PSPACE, this length-increasing hypothesis is indeed necessary.)

To establish an analogue of Ko, Long, and Du's lemma, we first show a version of Dowd's Theorem for strictly polynomial-space reductions.

THEOREM 3. *Every two strictly polynomial-space* 1*-equivalent sets are strictly polynomial-space isomorphic.*

Using Theorem 3 it is now straightforward to show

THEOREM 4. *If* P = PSPACE, *then every two* 1*-equivalent sets are p-isomorphic.*

Therefore, by combining Theorems 1, 2, and 4 we obtain our main result:

THEOREM 5. *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.*

One interesting feature of all of the work cited above is the central role played by Dedekind's construction for the Cantor-Bernstein Theorem. The constructions for Myhill's, Dowd's, and Berman and Hartmanis's Theorems as well as our Theorem 3 are all effective variants of the Dedekind's construction. The proofs of Ko, Long, and Du's Theorem and our Theorems 1 and 2 establish that certain plausible effective forms of Cantor-Bernstein fail if certain complexity classes separate.

The broader context. The results reported in this paper are a part of the body of research stemming from Berman and Hartmanis's *Isomorphism Conjecture* discussed in the beginning of this section. The bulk of that research concerns the possible structure of complete degrees of important complexity classes. For example, the complete m-degree of NEXP (nondeterministic exponential time) is known to consist of a single 1-degree [GH89], the complete m-degree of EXP is known to consist of a single 1-li-degree [Ber77], and, for the complete m-degrees of PSPACE and NP, no absolute results are known. Kurtz, Mahaney, and Royer's paper [KMR90] surveys this work through the late 1980s. Since the work of this paper does not particularly concern complete degrees, it is beyond our scope to update the survey [KMR90] to the present.

Acknowledgments. We wish to thank Per Brinch Hansen for the well-timed sarcastic remark that prompted us to finish the revision of this paper.

§2. Isomorphisms.

In this section we provide the proofs of our Theorem 3 and Dowd's Theorem. We also sketch the proofs of Myhill's and Berman and Hartmanis's Theorems. As mentioned above, the starting point for all these results is the standard proof of

THE CANTOR-BERNSTEIN THEOREM. *Given sets X and Y for which there are one-one functions $f: X \to Y$ and $g: Y \to X$, there is a one-to-one correspondence between X and Y.*

The theorem, as stated, concerns the category of sets, but it and its standard proof have many variants in other settings. The general setting for this paper is ***DP***, the category of decision problems, defined as follows.

DEFINITION 6. ***DP*** is the category with objects of the form (A, X), where X is a copy of the natural numbers and $A \subseteq X$, and with homomorphisms of the form $f: (A, X) \to (B, Y)$, where $f: X \to Y$ is a set-theoretic function with the additional property that, for all $x \in X$, $x \in A$ if and only if $f(x) \in B$.

DP in and of itself is not terribly interesting, but the subcategories of ***DP*** obtained by adding more requirements on homomorphisms, e.g., that each be polynomial-time computable, provide an adequate categorical setting for most work on strong reducibilities in recursion theory and complexity theory.

It is easily seen that a ***DP***-isomorphism is an $f: (A, X) \to (B, Y)$ such that $f: X \to Y$ is a one-to-one correspondence. So, the Cantor-Bernstein Theorem restated for ***DP*** is:

THEOREM 7. *Given (A, X) and (B, Y) in **DP** with $f: (A, X) \to (B, Y)$ and $g: (B, Y) \to (A, X)$ such that $f: X \to Y$ and $g: Y \to X$ are both one-one (i.e., monic), then (A, X) and (B, Y) are isomorphic.*

The argument we sketch for this theorem is essentially Dedekind's proof of the Cantor-Bernstein Theorem[5]. Before giving this proof, we state some general conventions that shall hold throughout the remainder of this paper.

CONVENTION 8. (a) Suppose A, B, X, Y, f, and g are as in the statement of Theorem 7. Without loss of generality, we assume that $X = \omega$ and $Y = \omega'$ where ω' is 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 $(\omega \cup \omega')$. Also, $\overline{A}$ and $\overline{B}$ respectively denote $\omega - A$ and $\omega' - B$.

(b) Let G be the directed graph $(\omega \cup \omega', E)$, where

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

G is clearly bipartite. Since $f: \omega \to \omega'$ and $g: \omega' \to \omega$ 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 *chains*. If a chain has a vertex of in-degree zero, we call this vertex the *root* of the chain. Each chain 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'.

Since f and g are ***DP***-homomorphisms, it follows that for a given chain C either (i) all of C's ω-vertices are in A and all of C's ω'-vertices are in B or else (ii) all of C's ω-vertices are in $\overline{A}$ and all of C's ω'-vertices are in $\overline{B}$.

(c) We say that a function $h: \omega \to \omega'$ *respects chains* if and only if for all x, x and $h(x)$ belong to the same chain. It follows by the properties of chains just noted that, if $h: \omega \to \omega'$ respects chains, then $h: (A, \omega) \to (B, \omega')$.

---

[5]For the history of this theorem and its proof, see either of Moore's [Moo82] or Ferreirós' [Fer99] excellent books. A summary of this history is given in Kurtz, Mahaney, and Royer's survey [KMR90]. As Ferrirós notes [Fer99, p. 240], the Cantor-Bernstein Theorem is an easy consequence of Theorem 63 in Dedekind's *Was sind und was solen die Zahlen?* [Ded88], but Dedekind seems never to have pointed this out to Cantor.

PROOF OF THEOREM 7 (after Dedekind [Ded88]). Define $\pi\colon \omega \to \omega'$ by

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

Note that π respects chains; hence, $\pi\colon (A,\omega) \to (B,\omega')$. Moreover, for each chain C, π gives a one-to-one correspondence between the collection of ω vertices of C and the collection of ω' vertices of C. (To see this, simply check that π works as claimed for each of the four possible structures of C.) Since the chains partition G, it follows that $\pi\colon \omega \to \omega'$ is a one-to-one correspondence. Hence, π is a ***DP***-isomorphism between (A,ω) and (B,ω'). ⊣

The proof of Berman and Hartmanis's Theorem builds directly on the above construction — the assumptions on f and g in the theorem provide sufficient conditions for π of (1) to be computable and invertible in polynomial time. Here are the details.

THEOREM 9 (Berman and Hartmanis's Theorem, Restated). *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 (after [BH77]). Suppose that f and g satisfy hypotheses (a), (b), and (c). Let π be as in (1). So, π is a ***DP***-isomorphism between (A,ω) and (B,ω'). 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 where 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, since f and g are both polynomial-time computable and p-invertible, it follows that π is also. ⊣

It is easily shown that there are recursive f and g for which π as defined in (1) fails to be computable. So we need a different construction for Myhill's Theorem.

THEOREM 10 (Myhill's Theorem, Restated). *Every two recursively* 1*-equivalent sets are recursively isomorphic.*

PROOF SKETCH (after [Myh55]). 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 ***DP***-isomorphism between (A,ω) and (B,ω'). ⊣

Our proof of Theorem 3 is in the spirit of the above argument, but in addition we must observe space bounds on the isomorphism being built, and thus our construction is considerably more delicate.

THEOREM 11 (Theorem 3, Restated). *Every two strictly polynomial-space* 1*-equivalent sets are strictly polynomial-space isomorphic.*

PROOF. Suppose f and g are one-one strictly polynomial-space computable functions. Below we describe the construction of $\tilde{\pi}$, a strictly polynomial-space computable isomorphism that respects chains. In the construction of the previous proof, although the root of a given 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 chains is more myopic; at each stage we can only see a portion of a 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 is matched with a vertex of roughly the same length.

Let the graph G be as above. 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_n (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 $h: \omega_n \to \omega'_n$ *respects n-chains* if and only if, for each $x \in \text{domain}(h)$, $h(x)$ is in the same n-chain as x.

Our construction of $\tilde{\pi}$ will be in stages. For each n, $\tilde{\pi}_n: \omega_n \to \omega'_n$ 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, one-one partial map between ω_n and ω'_n. We call the elements of $(\text{domain}(\tilde{\pi}_n) \cup \text{range}(\tilde{\pi}_n))$ the vertices *matched as of stage n*. Note that in order to be one-one *and* respect n-chains, it must be the case that biased n-chains (that have an odd number of elements) end up with at least one vertex that is unmatched as of stage n. In our construction, we 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 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 smallest 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 $z_1, z_2, \ldots, z_k$ 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

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

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 $\widetilde{\pi}_n$ is one-one, respects n-chains, and satisfies the invariant (2).

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

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

LEMMA 12. *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. So, $|z| \leq |\hat{z}|$. Since in z's $|\hat{z}|$-chain, z is followed by $\hat{z}$, a length $|\hat{z}|$ vertex, 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 13. *Both $\lambda n, x \in \omega_n . \widetilde{\pi}_n(x)$ and $\lambda n, y \in \omega'_n . \widetilde{\pi}_n^{-1}(y)$ are computable within $\mathscr{O}(n \cdot q(n))$ space.*

PROOF SKETCH. To compute $\widetilde{\pi}_n(x)$ using (3), one needs to

- compute $\widetilde{\pi}_{n-1}(x)$,
- if it is defined, output the result,
- if not, then x is one of the z_i's for x's n-chain, in that 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 preceding 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 $\widetilde{\pi}_{n-1}(x)$ 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 $\widetilde{\pi}_{n-1}$ 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 one-one strictly polynomial-space computable functions, it is clear that traversing an n-chain can be done in $\mathscr{O}(q(n))$ space. It is also clear that in using (3) to compute $\widetilde{\pi}_n(x)$, the depth of recursions is no more than n. Thus, it follows that $\widetilde{\pi}_n(x)$ can be computed within the required space bound. The argument for $\widetilde{\pi}_n^{-1}$ follows by symmetry. ⊣

Define $\widetilde{\pi} = \cup_{n\in\omega}\widetilde{\pi}_n$. Since each $\widetilde{\pi}_n$ extends $\widetilde{\pi}_{n-1}$, $\widetilde{\pi}$ is well defined. Since each $\widetilde{\pi}_n$ is one-one and respects n-chains, $\widetilde{\pi}$ is also one-one and respects chains. By Lemma 12, $\widetilde{\pi}$ is total and onto. By (3) and Lemma 12 we also have that, for all $x \in \omega$, $|\widetilde{\pi}(x)| \leq q(|x|)$ and $|x| \leq q(|\widetilde{\pi}(x)|)$. Finally, by Lemmas 12 and 13, we have that $\widetilde{\pi}$ and $\widetilde{\pi}^{-1}$ are both polynomial-space computable. ⊣ Theorem 3

THEOREM 14 (Dowd's Theorem [Dow82]). *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. This provides a somewhat crisper proof this theorem than Dowd's (unpublished) original.

In the proof of Lemma 13 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 15. *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 $C_{\hat{n}}$ that 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 the unmatched vertex of $C_{\hat{n}}$. ⊣

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

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

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

LEMMA 16. *Both* $\lambda n, x \in \omega_n.\widetilde{\pi}_n(x)$ *and* $\lambda n, y \in \omega'_n.\widetilde{\pi}_n^{-1}(y)$ *are computable in* $\mathscr{O}(q(n))$ *space.*

By Lemma 12 we have that $\widetilde{\pi} = \lambda x.\widetilde{\pi}_{q(|x|)}(x)$ and $\widetilde{\pi}^{-1} = \lambda x.\widetilde{\pi}^{-1}_{q(|x|)}(x)$. Hence, by Lemma 16,

COROLLARY 17. *Both* $\widetilde{\pi}$ *and* $\widetilde{\pi}^{-1}$ *are computable in* $\mathscr{O}(q(q(|x|)))$ *space.*

If f and g are one-one 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 17, the theorem follows. ⊣ Theorem 14

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, that focuses on the reductions themselves, we look closer at the structure of the chains formed by the 1-reductions, and in doing so, we obtain somewhat stronger results.

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 18. *Suppose two sets are (polynomial-time) 1-equivalent as witnessed by reductions f and g that 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 9. It isn't however. If f and g are such that there are no cyclic chains, then one can show that the hypotheses of Theorem 9 are equivalent to those of Theorem 18. We can use the construction for Theorem 3 to obtain a strictly stronger result than Theorems 9 and 18. In order to state this result we introduce two more technical notions.

We say that f is *honestly-invertible* if and only if 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.

We say that f and g's n-chains have *polynomial-time uniform extremities* if and only if there is a procedure that, given n and a $z \in (\omega_n \cup \omega_n')$, 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.

We can now state:

THEOREM 19. *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 3 in polynomial-time. This is straightforward and we omit the details.

Theorem 19's hypotheses are strictly weaker than those of Theorems 9 and 18 as shown by Proposition 33 below. Hypothesis (c) is still fairly strong, however.

It will be apparent from the proof of Theorem 1 in the next section that there are one-to-one, polynomial-time computable f and g such that the problem of finding just the tails of the corresponding n-chains is PSPACE-complete.

§3. Inequivalences. Our proofs of Theorems 1 and 2 follow the same general strategy as the proof of Ko, Long, and Du's Theorem. To lay out this strategy, we start by sketching a proof of that theorem after setting a few more conventions for the arguments to follow.

CONVENTION 20. (a) $\langle \cdot, \cdot \rangle$ denotes a polynomial-time computable and invertible pairing function such that $|\langle x, y \rangle| \in O(|x| + |y|)$. The pairing function in [Rog67] will do.

(b) We say that '$f(\vec{x})$ has a $\mathcal{P}oly(g(\vec{x}))$ bound' when there is a polynomial p such that for all $\vec{x}$, $f(\vec{x}) \leq p(g(\vec{x}))$. Similarly, we say that '$f(\vec{x})$ has a $2^{\mathcal{P}oly(g(\vec{x}))}$ bound' when there is a polynomial p such that for all $\vec{x}$, $f(\vec{x}) \leq 2^{p(g(\vec{x}))}$.

(c) We say that a function $h: \omega \to \omega'$ (or $h: \omega' \to \omega$) *crosses a chain* C if and only if for some x, a vertex of C, $h(x)$ fails to be a vertex of C.

THEOREM 21 (Ko, Long, and Du's Theorem, Restated). *Suppose that* P $\neq$ UP. *Then there exist* 1*-li equivalent sets that are incomparable with respect to p-invertible reductions. Moreover, there are such sets that are* 2*-tt complete for* EXP.

PROOF SKETCH. Since we are assuming P $\neq$ UP, by Proposition 2.1 of [KLD87], there exists a length-increasing one-way function t. Define $f: \omega \to \omega'$ 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 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) $f: A \leq^{\mathrm{p}}_{\text{1-li}} B$ and $g: B \leq^{\mathrm{p}}_{\text{1-li}} A$,

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

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

The diagonalization depends on the following key lemma.

LEMMA 22 (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.*

PROOF. We handle the case of $h: \omega \to \omega'$. 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

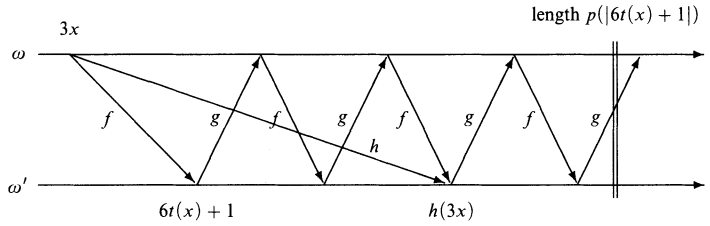

FIGURE 1. $h(3x)$ lands in V_y.

would 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. ⊣ Theorem 22

Returning to the proof of Theorem 21, 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 $h: \omega \to \omega'$ 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 22, one can construct A and B satisfying (6) and (8) by a elementary, noneffective diagonalization: start with all 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 A and B that satisfy (7) in addition to (6) and (8), a more delicate construction is needed. We handle this construction by means of a general technical lemma that is also used in the proofs of Theorems 1 and 2 below. To state this lemma, we introduce the following terminology. Suppose C is a chain with root r. The *i-th 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 i-th successor of r for some $i \leq |r|$, (b) for each $j \leq i$, the j-th 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 this paper's appendix.

LEMMA 23 (The Chain Painting Lemma). *Suppose the following*:

1. *$f: \omega \to \omega'$ and $g: \omega' \to \omega$ are one-one and polynomial-time computable.*
2. *$r: \omega \to (\omega \cup \omega')$ is one-one, $2^{\mathcal{P}oly(n)}$-time computable, and, for each x, $r(x)$ is the root of a 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. *$s: \omega \to \omega$ is polynomial-time computable, and for all distinct $x, y \in \omega$, $s(x)$ and $s(y)$ are in chains distinct from all the $\widehat{C}_z$'s and from each other. For each x, let $\widehat{D}_x$ denote $s(x)$'s chain.*
5. *Given $z \in (\omega \cup \omega')$ and $x \in \omega$, deciding whether z is a vertex of $\widehat{C}_x$ can be done in $\mathscr{P}oly(|z| + x)$-time (or equivalently, in $\mathscr{P}oly(|z| + 2^{|x|})$-time).*
6. *Given $z \in (\omega \cup \omega')$, deciding whether z is in one of the $\widehat{D}_y$'s, and, if so, which y, all can be done in $\mathscr{P}oly(|z|)$-time.*

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

(a) *$f: A \leq_1^{\mathrm{p}} B$ and $g: B \leq_1^{\mathrm{p}} A$,*
(b) *A and B are 2-tt complete for* EXP, *and*
(c) *there is no polynomial-time computable $h: \omega \to \omega'$ (respectively, $h: \omega' \to \omega$) that 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 23, they are easily satisfied in each of our applications of the lemma. In the context of the proof of the present theorem:

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

$q = \lambda n.[n + 1]$; and $s = \lambda x.[6x + 2]$. Lemma 22 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 23. ⊣ Theorem 21

We now apply the technique used in the proof above to 1-reductions that are not necessarily length-increasing. With the (most likely) weaker assumption that P ≠ PSPACE, we obtain two different inequivalences. The first of these (Theorem 24) involves honest m-reductions; the other (Theorem 27) concerns p-invertible reductions and uses the same basic plan with one additional twist.

THEOREM 24 (Theorem 1, Restated). *Suppose that* P ≠ PSPACE. *Then there exist 1-equivalent sets that are incomparable with respect to honest m-reductions. Moreover, there are such sets that 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 21. 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 chains; then, by an application of the Chain Painting Lemma, we produce the two sets required by the theorem. In Theorem 21's proof, the 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 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 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 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 $\tau\colon Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, R, N\}^k$. 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 $\sigma\colon Q \times \Sigma^k \to Q \times \Sigma^k \times \{L, R, N\}^k$ 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 25. *Suppose M is a multi-tape Turing machine that computes a function $t\colon \omega \to \omega$ and that runs in space $S(n)$. Then, there there is an $\mathscr{O}(S(n)^2)$ 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 *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. that 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 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 chains. Let $\#\colon \mathrm{ID} \to \omega$ be a one-one, onto function, and such that

- the set $\#(\widehat{\mathrm{ID}})$ is polynomial-time decidable;
- 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 $\mathscr{P}oly(|i|)$-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 $f : \omega \to \omega'$

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 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 a $C'_{x,v}$ chain is analogous.

LEMMA 26 (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 $h\colon \omega \to \omega'$ case. The $\omega' \to \omega$ 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 $|\mathit{start}(x,\mathbf{0}^v)| \in \Omega(|x| + 2^{|v|})$. Thus, for each x and all sufficiently large v,

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

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 $h(\mathit{start}(x,\mathbf{0}^v))$ is in $C_{x,v}$. Then, for some z, $h(\mathit{start}(x,\mathbf{0}^v)) = \mathit{idle}(x,v,z,I)$, where I is the final i.d. of M on input x.

PROOF OF CLAIM. Since *start* is increasing in both arguments and since $x > x_0$, we have by our choice of k and x_0 that $|\mathit{start}(x,\mathbf{0}^v)|^{1/k} < |h(\mathit{start}(x,\mathbf{0}^v))|$. 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 $|h(\mathit{start}(x,\mathbf{0}^v))| > p_L(|x|+|v|)$, $h(\mathit{start}(x,\mathbf{0}^v))$ cannot be in the active part of $C_{x,v}$. Thus, since $h(\mathit{start}(x,\mathbf{0}^v))$ is in $C_{x,v}$, it must be in the idle part of $C_{x,v}$. Therefore, the claim follows. ⊣

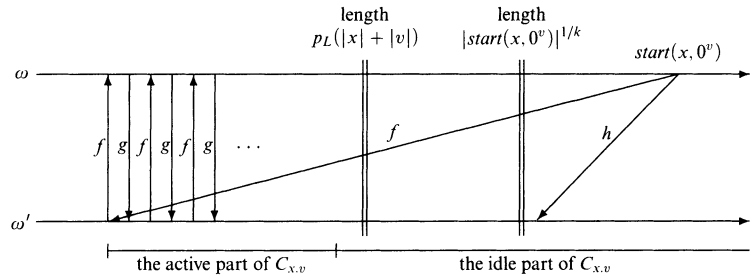

FIGURE 2. $h(start(x, \mathbf{0}^v))$ lands in $C_{x,v}$.

Suppose by way of contradiction that the lemma is false. So, for all but finitely many x, $h(start(x, \mathbf{0}^{p_*(|x|)}))$ is in $C_{x,p_*(|x|)}$. Then by the claim, for all but finitely many x, one can determine $L(x)$ by: (i) computing $h(start(x, \mathbf{0}^{p_*(|x|)}))$, (ii) from this value extracting the final i.d. of M on input x, and (iii) from this i.d. determine $L(x)$. 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 26

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 $s = \lambda x.start(x, \mathbf{1})$. 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 that lemma there are sets A and B that are 1-equivalent, 2-tt complete for EXP, but that are not honest m-comparable. ⊣ Theorem 24

We now turn to the second of our two main inequivalences. In the proof of the prior theorem we had, under the assumption of P $\neq$ PSPACE, that no polynomial-time honest equivalence (not even one-one) could be substituted for an unrestricted polynomial-time 1-equivalence. Here we show, again under the the assumption of P $\neq$ PSPACE, the more fine-grained result that a no p-isomorphism can 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 9 have that is not required here is that of being length-increasing. Thus if P $\neq$ PSPACE, the length-increasing requirement of Theorem 9 is necessary.

THEOREM 27 (Theorem 2, Restated). *Suppose that* P $\neq$ PSPACE. *Then there exist p-invertible equivalent sets that fail to be p-isomorphic. Moreover, there are such sets that are* 2*-tt complete for* EXP.

Our proof of this theorem will run along lines similar to our argument for Theorem 24. In particular, the chains we construct will look similar to those of Theorem 24, 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 for Theorem 24 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 as 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 ramping down to the start of the active region; there 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 one) with a vertex in the idle region, thus revealing the result of the computation as in our proof of Theorem 24, 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." The function 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 comes in selecting what maps, h, to diagonalize against. For Theorem 24, 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 need to consider all possible pairs of polynomial-time functions, on the suspicion that any pair may represent a p-isomorphism and its inverse.

Before beginning the proof, we establish a few conventions regarding universal functions.

CONVENTION 28. (a) Henceforth, $\langle \varphi_i \rangle_{i \in N}$ denotes 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

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

is computable in $\mathcal{O}((|i| + |x| + n)^2)$ time.

(b) For each k, ℓ, and x, define

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

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

PROOF OF THEOREM 27. Let L be an element of (PSPACE − P). As noted in the proof of Theorem 24, there is a reversible Turing machine, M, that computes $\lambda x . \langle x, L(x)\rangle$ in polynomial space.

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

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

To encode M-computations into chains, we use essentially the same tools developed in the proof of Theorem 24. 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) $\quad \{\, y : d(x, i, \mathbf{0}^y) = 0 \,\}$ is a nonempty, *finite* initial segment of ω.

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

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

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

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

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 chains other than the $C_{x,i}$'s.

A $C_{x,i}$ chain has the following structure, partly 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 the largest number such that $d(x, i, \mathbf{0}^y) = 0$. Then the chain "ramps" down from

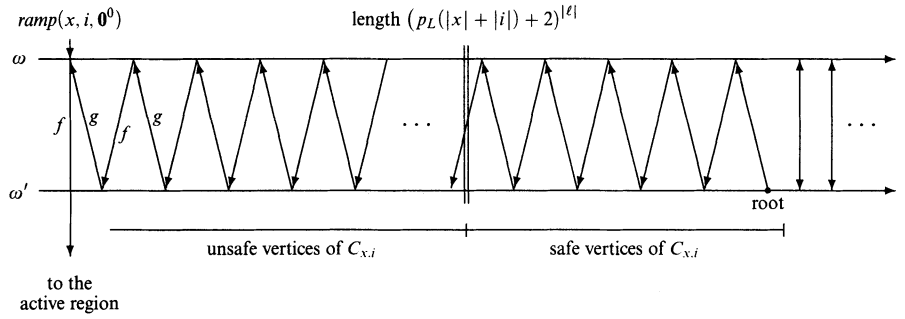

FIGURE 3. The ramp portion of $C_{x,i}$.

$ramp(x, i, \mathbf{0}^y)$ to $ramp(x, i, \mathbf{0}^{y-1})$ and then to $ramp(x, i, \mathbf{0}^{y-2})$ and so on until it arrives at $ramp(x, i, \mathbf{0}^0) \in \omega$. Note that by the definitions of f and g, each $C_{x,i}$ vertex of the form $ramp(x, i, \mathbf{0}^y)$ is in ω precisely when y is even. Also note that by the definition of *ramp*, as y decreases, so does the *length* of $ramp(x, i, \mathbf{0}^y)$. Returning to our tour of $C_{x,i}$, the vertex $ramp(x, i, \mathbf{0}^0) \in \omega$ is followed by the vertex $active(x, i, initial(x)) \in \omega'$. Then, as in the previous proof, f and g conspire to simulate M in input x—successive active 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 $h: \omega \to \omega'$ is a chain-respecting p-isomorphism. Fix x and fix an i such that $i = \langle j, k, \ell \rangle$, $\psi_j^\ell = h$, and $\psi_k^\ell = h^{-1}$. 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 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 below. Once we know such a v exists, we can compute $L(x)$ as in Theorem 24 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:

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

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

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

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

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

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 that 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, with $i = \langle j,k,\ell \rangle$, we have that

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

and the least ramp vertex of $C_{x,i}$ that is of length $\geq bnd(x,i)$ is an ω-vertex. This last property of *bnd* helps to simplify the definition of d and the proof of Lemma 30 below.

LEMMA 29. *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*

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

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 $W'_{x,i}$:

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

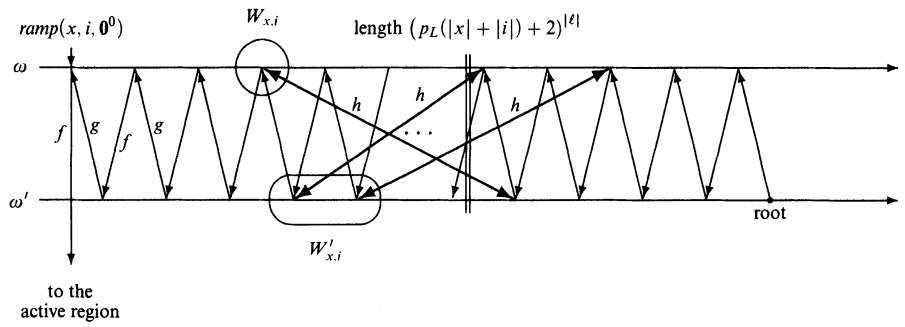

FIGURE 4. The root is chosen to yield one more unmatched safe ω'-vertex.

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

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

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

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 one-one correspondence between $V_{x,i} - h^{-1}(W'_{x,i})$ and $V'_{x,i} - h(W_{x,i})$, and thus

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

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 $C_{x,i}$. ⊣

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

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

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

In defining d we have to worry about the time cost of determining $\|W_{x,i}\|$ and $\|W'_{x,i}\|$. To help in bounding this cost, define

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

Since the number of ramp vertices of $C_{x,i}$ of length $< bnd(x,i)$ is no more than $bnd(x,i)$ and since $\lambda k,\ell,y.\psi_k^\ell(y)$ is computable in $\mathscr{O}(\,(|k|+3|y|)^{2|\ell|}\,)$ time, it follows that one can test whether $\|W_{x,i}\| = \|W'_{x,i}\|$ in $\mathscr{O}(t(x,i))$ 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 30 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 $\mathscr{P}oly(y+|x|+|i|)$ time by: running the computation of $t(x,i)$ for $p_\star(y)$ steps and, if the computation fails to halt within y steps, then we know that $y < t(x,i)$, and if the computation does halts within $p_\star(y)$ steps, we can do the comparison within $p_\star(y)$ steps.

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 30. *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:

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

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

Therefore, by (14) we have:

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

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 31 (The Chain Crossing Lemma). *Suppose $h\colon \omega \to \omega'$ 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) $\lambda x.ramp(x, i, \mathbf{0}^0)$ 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 $ramp(x, i, \mathbf{0}^y)$ 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 so h respects chains almost everywhere. Then by Lemmas 29 and 30, 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 $idle(x, i, z, I)$ be this idle vertex.
2. From $idle(x, i, z, I)$ 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 the proof of Theorem 24. 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 $s = \lambda x.ramp(x, 0, \mathbf{1})$. 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 that lemma there exist sets A and B that are p-invertible 1-equivalent, 2-tt complete for EXP, but that are not p-isomorphic. ⊣ Theorem 27

We can use the analysis of the proofs of the previous two theorem to show two more inequivalences, one for one-one polynomial-space reductions and another for fairly strong polynomial-time reductions.

THEOREM 32. *There are polynomial-space* 1*-equivalent sets that are not polynomial-space isomorphic.*

PROOF SKETCH. We again follow the plan of the previous proofs: We construct one-one polynomial-space computable functions f and g; prove that every honest polynomial-space computable function must promptly cross infinitely many of a particular collection of 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. Define $f : \omega \to \omega'$ by:

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

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 one-one and polynomial-space computable. Given any fixed y, let $n = 2^y + 1$. The functions f and g give rise to the following 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 $h: \omega \to \omega'$ is a polynomial-space isomorphism that respects 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), that 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 22, we can define the two desired sets. ⊣

PROPOSITION 33. *There are sets A and B that 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) *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 $f: \omega \to \omega'$ by the two following equations.

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

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 vertices in $\mathbf{1}^*$.

Now, suppose that $h: \omega \to \omega'$ is one-one and length-increasing. If h respects 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 computable. Thus any such polynomial-time computable h must cross infinitely many chains. So, we are done by the remarks following the proof of Lemma 22. ⊣

With a bit more work we could obtain A and B as above that are also 2-tt complete for EXP.

Appendix. Proof of the chain painting lemma. Recall that, for an 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 $\leq |r|$.

For all j, l, and x, define $\widetilde{\psi}_{\langle j,l\rangle}(x) = T(j, x, (|x|+2)^{\log|l|})$, where T is as in Convention 28(a). Using the definition of T and the time bound of Convention 28(a) it is straightforward to argue that $\langle\widetilde{\psi}_i\rangle_{i\in\omega}$ is an enumeration of the polynomial-time computable functions and that, given i and x, $\widetilde{\psi}_i(x)$ is computable within $2^{p(\log(|i|+|x|))}$ time for some polynomial p. To handle maps both from ω to ω' and from ω' to ω, we define, for all i:

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

LEMMA 34 (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 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 distinct* $y, z \in \omega$, $s(y)$ *and* $s(z)$ *are in 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) $f\colon A \leq_1^{\mathrm{p}} B$ *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* C_x*'s and that* $\leq_{\mathrm{m}}^{\mathrm{p}}$*-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 22, 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 chains of the form C_k or D_k unpainted and unreserved, all the rest of the chains painted *green*, and all $i \in \omega$ uncanceled. 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 uncanceled $i \leq k$, 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 uncanceled.
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 $A = \{ x \in \omega : x\text{'s chain is blue} \}$ and $B = \{ y \in \omega' : y\text{'s chain is blue} \}$. It is immediate that $f : A \leq_1^p B$ and $g : B \leq_1^p A$.

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 uncanceled 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 uncanceled 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 uncanceled 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 uncanceled 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 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 it follows that $|x_k|$ 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 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 uncanceled,
  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 $\mathcal{P}oly(k) = 2^{\mathcal{P}(|k|)}$. 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:
  (i) Detecting whether ψ_i promptly crosses C_k can be done within $2^{\mathcal{P}oly(|i|+|k|)}$-time.
  (ii) Finding x_k can be done within $2^{\mathcal{P}oly(|k|)}$-time.

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

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

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. ⊣

Conclusion (a) of the lemma holds as mentioned above. Claims 3 and 4 prove (b). Conclusion (c) follows from Claim 1. ⊣ Lemma 34

REFERENCES

[Ben89] C. BENNETT, *Time/space trade-offs for reversible computation*, ***SIAM Journal on Computing***, vol. 18 (1989), pp. 766–776.

[Ber77] L. BERMAN, *Polynomial reducibilities and complete sets*, ***Ph.D. thesis***, Cornell University, 1977.

[BH77] L. BERMAN and J. HARTMANIS, *On isomorphism and density of NP and other complete sets*, ***SIAM Journal on Computing***, vol. 1 (1977), pp. 305–322.

[Ded88] R. DEDEKIND, ***Was sind und was sollen die Zahlen?***, Vieweg, 1888, English translation in [Ewa96], Vol. 2.

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

[Ewa96] W. Ewald (editor), ***From Kant to Hilbert: A sourcebook in the foundations of mathematics***, Oxford University Press, 1996, in two volumes.

[Fer99] J. FERREIRÓS, ***Labyrinth of thought: A history of set theory and its role in modern mathematics***, Birkhäuser, 1999.

[GH89] K. GANESAN and S. HOMER, *Complete problems and strong polynomial reducibilities*, ***Proceedings of the symposium on theoretical aspects of computer science***, Springer-Verlag, 1989, pp. 240–250.

[GJ79] M. GAREY and D. JOHNSON, ***Computers and intractability***, W. H. Freeman and Company, 1979.

[GS84] J. GROLLMANN and A. SELMAN, *Complexity measures for public-key cryptosystems*, ***Proceedings of the 25th annual ieee symposium on foundations of computer science***, IEEE Computer Society, 1984, pp. 495–503.

[GS88] ———, *Complexity measures for public-key cryptosystems*, ***SIAM Journal on Computing***, vol. 17 (1988), pp. 309–335.

[HKR93] S. HOMER, S. KURTZ, and J. ROYER, *On many-one and 1-truth-table complete sets*, ***Theoretical Computer Science***, vol. 115 (1993), pp. 383–389.

[Ko85] K. KO, *On some natural complete operators*, ***Theoretical Computer Science***, vol. 37 (1985), pp. 1–30.

[KLD87] K. KO, T. LONG, and D. DU, *A note on one-way functions and polynomial-time isomorphisms*, ***Theoretical Computer Science***, vol. 47 (1987), pp. 263–276.

[KMR90] S. KURTZ, S. MAHANEY, and J. ROYER, *The structure of complete degrees*, ***Complexity theory retrospective*** (A. Selman, editor), Springer-Verlag, 1990, pp. 108–146.

[Moo82] G. MOORE, ***Zermelo's axiom of choice: Its origins, development, and influence***, Springer-Verlag, 1982.

[Myh55] J. MYHILL, *Creative sets*, ***Zeitschrift für Mathematische Logik und Grundlagen der Mathematik***, vol. 1 (1955), pp. 97–108.

[Rog67] H. ROGERS, ***Theory of recursive functions and effective computability***, McGraw-Hill, 1967, Reprinted. MIT Press. 1987.

DEPT. OF COMPUTER SCIENCE
UNIVERSITY OF SOUTH CAROLINA
COLUMBIA, SC 29208, USA
E-mail: fenner@cs.sc.edu

DEPT. OF COMPUTER SCIENCE
UNIVERSITY OF CHICAGO
1100 E. 58TH ST., CHICAGO, IL 60637-1581, USA
E-mail: stuart@cs.uchicago.edu

DEPT. OF ELEC. ENG. AND COMPUTER SCIENCE
SYRACUSE UNIVERSITY
SYRACUSE, NY 13244, USA
E-mail: royer@ecs.syr.edu