



# Polynomial-Time Axioms of Choice and Polynomial-Time Cardinality

Joshua A. Grochow<sup>1</sup>

Accepted: 20 January 2023 / Published online: 15 May 2023  
© The Author(s) 2023

## Abstract

There is no single canonical polynomial-time version of the Axiom of Choice (AC); several statements of AC that are equivalent in Zermelo-Fraenkel (ZF) set theory are already inequivalent from a constructive point of view, and are similarly inequivalent from a complexity-theoretic point of view. In this paper we show that many classical formulations of AC, when restricted to polynomial time in natural ways, are equivalent to standard complexity-theoretic hypotheses, including several that were of interest to Selman. This provides a unified view of these hypotheses, and we hope provides additional motivation for studying some of the lesser-known hypotheses that appear here. Additionally, because several classical forms of AC are formulated in terms of cardinals, we develop a theory of polynomial-time cardinality. Nerode & Rempel (*Contemp. Math.* 106, 1990 and Springer Lec. Notes Math. 1432, 1990) developed a related theory, but restricted to unary sets. Downey (Math. Reviews MR1071525) suggested that such a theory over larger alphabets could have interesting connections to more standard complexity questions, and we illustrate some of those connections here. The connections between AC, cardinality, and complexity questions also allow us to highlight some of Selman’s work. We hope this paper is more of a beginning than an end, introducing new concepts and raising many new questions, ripe for further research.

**Keywords** Structural complexity · Function complexity · Axiom of choice · Cardinality · Function invertibility ·  $p$ -isomorphism · Structural properties of languages

---

Dedicated to the memory of Alan L. Selman

This article belongs to the Topical Collection: *Commemorative Issue for Alan L. Selman*  
Guest Editors: Mitsunori Ogihara, Elvira Mayordomo, Atri Rudra

✉ Joshua A. Grochow  
jgrochow@colorado.edu

<sup>1</sup> University of Colorado Boulder, Boulder, CO, USA

## 1 Introduction

“The Axiom of Choice is obviously true, the well-ordering principle is obviously false, and who can tell about Zorn’s Lemma?” —Jerry L. Bona, [55, p. 145]

The Axiom of Choice (AC), in one of its many equivalent incarnations, says that for every collection  $(X_i)_{i \in I}$  of non-empty sets  $X_i$  (where  $I$  is an arbitrary index set), there is a “choice function”  $x: I \rightarrow \bigcup_{i \in I} X_i$  such that  $x(i) \in X_i$  for all  $i$ . Despite being non-constructive, this axiom is extremely useful throughout much of mathematics. In classical, Zermelo–Fraenkel set theory (ZF), there are many statements that are equivalent to AC. We refer to the books by Herrlich [37] and Rubin & Rubin [53, 54] for the classical AC and its many equivalent versions. Yet despite being equivalent in ZF, many versions often “feel” different, as captured nicely by Bona in the above quotation.

In this paper, we study several statements that are classically equivalent to AC and propose polynomial-time analogues that seem not to be equivalent to one another (under standard complexity assumptions). We show their relationship with some standard—and some less standard—complexity questions, most with relations to Alan Selman’s work, which is part of why we thought it fitting to submit to this memorial volume.

**Informal Definition.** A *polynomial-time axiom of choice* is any statement  $S$  which, if the words “polynomial-time,” “polynomially-bounded,” and similar were removed from  $S$ , would be equivalent (in Zermelo–Fraenkel set theory) to the usual Axiom of Choice.

Throughout this paper, our general philosophy is that “set” should become “language in  $\mathbf{P}$ ” and “function” should become “function in  $\mathbf{FP}$ .” Where there are several obvious choices for how to formulate a polynomial-time version of AC equivalent to a given standard formulation, we generally avoid formulations that are trivially true or trivially false.

Because several statements classically equivalent to AC are phrased in terms of cardinals, we also develop a polynomial-time theory of “p-cardinals” in Section 4. The basic idea is as follows: Cantor’s idea of cardinality was that two sets have the same “size” iff there is a bijection between them, so by analogy, we say two languages  $A \subseteq \Sigma^*$ ,  $B \subseteq \Gamma^*$  should have the same p-cardinality if there is a polynomial-time computable, polynomial-time invertible bijection between  $A$  and  $B$ . The key difference between this and the notion of p-isomorphism is that a p-isomorphism  $A \rightarrow B$  must be given by a *total* p-computable, p-invertible bijection  $f: \Sigma^* \rightarrow \Gamma^*$  such that  $f(A) = B$ , whereas in our case we focus more on the sets themselves, only requiring our bijection to be computable and invertible in polynomial time *when restricted to the sets  $A, B$* . Outside these sets it need not be injective nor surjective.

Our notion of p-cardinality turns out to be related to several complexity-theoretic notions of interest in Selman’s work, including p-enumerability (Observation 4.15 and Proposition 4.17), the Isomorphism Conjecture (Section 4.2), p-rankability (Section 4.3), and immunity (in Propositions 4.6 and 4.36). We suspect it has further

relationships with other of Selman’s interests including disjoint NP pairs [17, 22–24], mitoticity [18–20], splitting [21], and p-selectivity [34, 48, 58, 59], but leave these as exciting future directions.

## 1.1 Related Work

Constructive versions of AC, even in the context of computability, have been studied previously, e. g., [8, 13], but we are not aware of other work studying polynomial-time analogues of AC.

For p-cardinality, the most closely related work is that of Nerode & Remmel [49–52]. They developed not only a theory of polynomial-time cardinals, called “polynomial equivalence types” or “PETs” (in analogy with previously studied recursive equivalence types [11], see [10]), but also began to develop a theory of polynomial-time structures such as vector spaces, groups, etc., in analogy with work on computable structures (see, e. g., [3, 9, 47]). However, their polynomial-time theory was developed for unary languages.

While the choice to use unary languages enabled many analogies with the recursive case to go through and led to a rich and interesting theory, it was mostly disconnected from more standard complexity questions such as P versus NP, because the latter are typically formulated over an alphabet of size at least 2. Downey, in his review [12] of [51] writes:

“What is not yet clear is the relationship, if any, of complexity questions (such as  $P \stackrel{?}{=} NP$ ) with [51]. For instance, it would be truly fascinating if the techniques could be used to prove  $P = NP$  to be independent of [intuitionistic Zermelo–Fraenkel set theory] via an analogue of McCarty’s work [44, 45]. Perhaps what is needed is the development of the theory over the standard languages  $\subseteq \{0, 1\}^*$ .”—R. Downey [12]

Here (Section 4) we pursue one such development of the theory over standard languages, that is, over alphabets of size at least 2. While we do not yet realize Downey’s suggestion about independence from IZF, we lay some possible foundations of such a theory. In fact, this paper originally was focused on AC, with p-cardinality as just a side note, and it was our discovery of Downey’s quote that spurred us on to develop some of the foundations of p-cardinality further.

## 1.2 Selman’s Influence on my Work

I only met Alan personally a few times—I think once when I visited Steve Homer at Boston University (to give my first talk on complexity theory research), and once or twice at the Conference on Computational Complexity. But even from those few meetings one could tell that he was nice, generous with his ideas and time, had broad interests, and seemed genuinely very happy to talk about complexity theory.

But his influence on my work extends far beyond just our few chance meetings. The general theme is this: when I started graduate school and was talking with Lance

Fortnow and other members of the Theory group at U. Chicago (at the time: Stuart Kurtz, Ketan Mulmuley, Janos Simon, Laci Babai, and shortly after I joined, also Sasha Razborov), I was just over-the-moon with structural complexity theory. Properties of complexity classes, properties of (formal) languages, p-isomorphism, complexity cores, p-selectivity—I loved it all (still do!). Early on in my graduate career Selman’s two edited volumes [36, 61] were very influential on my thinking. Every chapter of [61] is related to something I have worked on at one point in my life (even if not resulting in any publications). They are still some of my favorite advanced complexity books on my shelf—and yes, I actually have the physical books, on a physical bookshelf.

While working on my first complexity paper [16], Lance pointed me to Selman’s taxonomy of function classes [62–64], and one can see that NPMV and its kin play an important role in that paper. Personal admission: I never really understood the definition of TFNP until reading Selman’s work on NPMV, NPSV, and their relatives. One of Selman’s results was in fact part of the motivation for [16] (discussed in my master’s thesis [28, Section 3.2]): although most function classes behave like their decision class counterparts, we have  $\text{P}^{\text{NP}[\log]} = \text{P}_{\text{tt}}^{\text{NP}}$  [7, 31, 67] but Selman showed that for the corresponding function classes this is unlikely:

**Theorem 1.1** (Selman [63])  $\text{FP}^{\text{NP}[\log]} = \text{FP}_{\text{tt}}^{\text{NP}}$  implies  $\text{NP} = \text{RP}$  and  $\text{P} = \text{UP}$ .

After that first complexity paper, I was hooked (if I wasn’t before). One of my favorite open oracle questions is still from Selman’s work. It remains open whether  $\text{NP} = \text{UP}$  collapses PH. But more than that, we don’t even know that it requires non-relativizing techniques. We wish to highlight this question here:

**Open Question 1.2** (Folklore) *Build an oracle relative to which  $\text{NP} = \text{UP}$  but PH is infinite.*

Hemaspaandra, Naik, Ogihara, and Selman [35] showed that a closely related statement does in fact collapse PH to  $\Sigma_2\text{P}$ . Namely, they showed that this collapse of PH follows from the assumption that  $\text{NPMV}_g \subseteq_c \text{NPSV}_g$ . All terms are defined in Section 2, but for now, we can say that the latter statement is equivalent to: for every language  $A \subseteq \Sigma^* \times \Sigma^*$  in P, there is another language  $B$  “refining”  $A$ , in the sense that for all  $x \in \Sigma^*$ , if  $(x, y) \in A$ , then there exists some  $y_x$  such that  $(x, y_x) \in A$  and  $(x, y_x)$  is the unique element of  $B$  with first coordinate  $x$ . This implies  $\text{NP} = \text{UP}$ , and feels philosophically very closely related (indeed, it is the kind of statement that someone not familiar with it might mistake for the definition of  $\text{NP} = \text{UP}$ ). I spent at least a summer, possibly more, in graduate school under Lance’s guidance trying to answer Question 1.2. The issue is that most naive ways of thinking about how to get  $\text{NP} = \text{UP}$ —and I was definitely naive at the time, probably still am—also imply  $\text{NPMV}_g \subseteq_c \text{NPSV}_g$ , and thus that  $\text{PH} = \Sigma_2\text{P}$ . To get around this barrier, we first sought to build an oracle relative to which  $\text{NP} = \text{UP}$  but  $\text{PH} \neq \Sigma_2\text{P}$ . We did not manage to do that, but I mention this because it led us to another of Selman’s works. Rather than trying to fully separate the levels of PH, which seems to require using the full strength of  $\text{AC}^0$  circuit lower bounds, we just sought to separate its second level,

and Baker and Selman had built an oracle separating just two levels of PH [4], so we looked to their work for inspiration.

At this point I don’t remember why, but around the same time I was also led to read Selman’s work on disjoint pairs, mitoticity, immunity, and p-selectivity (e. g., [19, 22, 30, 33, 48]). In fact, Selman’s work on search-to-decision reductions was part of what spurred my interest, leading me to discuss them in the context of Group Isomorphism with Youming Qiao in graduate school over a decade ago; we just recently made progress on that problem [29].

Even though we didn’t solve Question 1.2, it was a glorious several months being fully immersed in Selman’s work, that had an influence on my thinking and my career, and will stick with me for the rest of my life. Thanks Alan—your intellectual legacy lives on, and you are missed.

## 2 Preliminaries

Throughout,  $\Sigma$  (and sometimes also  $\Gamma$ ) is a finite alphabet of size at least 2, containing at least the symbols 0,1.  $\epsilon$  denotes the empty string, the unique string of length zero.  $|x|$  denotes the length of a string. Length-lexicographic ordering is defined by  $x <_{lex} y$  if  $|x| < |y|$ , or if  $|x| = |y|$  and  $x$  lexicographically precedes  $y$ , where lexicographic ordering is given according to some fixed (but unspecified, implicit) total order on  $\Sigma$ .  $\Sigma^*$  denotes the set of all finite strings over the alphabet  $\Sigma$ .

We move back and forth freely between  $\Sigma^*$  and  $\mathbb{N}$  as follows. The natural number associated to  $x \in \Sigma^*$  is the number of strings that are  $\leq_{lex} x$ , and vice versa, so e. g., if  $\Sigma = \{0, 1, 2\}$  we have the association

$$\begin{array}{cccccccccccccccccccc} 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & \dots \\ \epsilon & 0 & 1 & 2 & 00 & 01 & 02 & 10 & 11 & 12 & 20 & 21 & 22 & 000 & 001 & 002 & 010 & 011 & \dots \end{array}$$

For two strings  $x, y$  and a natural number  $n$ , when we write  $x + y$  or  $x + n$ , we first convert all strings to natural numbers using the above convention, then add them, and then convert back to the corresponding string, if needed.

For a language  $A \subseteq \Sigma^*$  we abuse notation by writing  $A(x) = 1$  iff  $x \in A$  and  $A(x) = 0$  iff  $x \notin A$ .  $\bar{A}$  denotes the complement of  $A$ ,  $\bar{A} = \{x \in \Sigma^* : x \notin A\}$ .

A polynomial-time many-one reduction from  $A$  to  $B$  is a p-computable total function  $f$  such that  $x \in A \Leftrightarrow f(x) \in B$  for all  $x \in \Sigma^*$ , equivalently,  $A(x) = B(f(x))$ . In this case we write  $A \leq_m^p B$ . If  $A \leq_m^p B$  and  $B \leq_m^p A$  we write  $A \equiv_m^p B$  and say that  $A$  and  $B$  have the same polynomial-time many-one degree (of difficulty). A bounded truth-table, or  $\leq_{btt}^p$ , reduction is a nonadaptive Turing reduction that makes  $O(1)$  queries (we only use  $\leq_{btt}^p$  in reference to other people’s results).  $A \subseteq \Sigma^*$  and  $B \subseteq \Gamma^*$  are p-isomorphic, denoted  $A \cong^p B$ , if there is a total, polynomial-time computable bijection  $f: \Sigma^* \rightarrow \Gamma^*$  for which  $f^{-1}$  is also computable in polynomial time, and such that  $f(A) = B$ .

For two languages  $A, B \subseteq \Sigma^*$ ,  $A \oplus B$  denotes the “tagged” disjoint union  $\{0a : a \in A\} \cup \{1b : b \in B\}$  and  $A \times B$  denotes the usual Cartesian product. Pairs, tuples, etc. of strings are encoded using any of the standard polynomial-time computable

and polynomial-time invertible bijections  $\Sigma^* \rightarrow \Sigma^* \times \Sigma^*$ . We write  $A =^* B$  if their symmetric difference  $(A \setminus B) \cup (B \setminus A)$  is finite.

We follow notation for partial, multi-valued functions set down by Selman [35, 63]. A (possibly) partial, multivalued function  $f$  is technically a relation  $\subseteq \Sigma^* \times \Sigma^*$ , but we think of the first coordinate of this relation as inputs and the second as outputs, and we use standard functional language and notation. Rather than the relation notation  $xfy$ , we write  $f(x) \mapsto y$ , but note that there may be more than one such  $y$  for a given  $x$  (so we are careful *not* to write  $f(x) = y$  unless there is exactly one output for  $f(x)$ ). We write

$$\text{set-}f(x) = \{y : f(x) \mapsto y\} \quad \text{and} \quad \text{dom}(f) = \{x : \text{set-}f(x) \neq \emptyset\}.$$

Nondeterministic Turing machine transducers—machines with separate output tapes—naturally compute partial, multi-valued functions. Given such a machine  $M$ , we define  $f_M(x) \mapsto y$  iff there is some accepting computation of  $M$  that halts with  $y$  on its output tape, and we say that  $f_M$  is computed by  $M$ . There is a related but subtly different function also associated with  $M$  that will be useful, namely  $\text{acc}_M(x) \mapsto y$  iff  $y$  is the ordered list of nondeterministic choices made on an accepting branch of  $M$ ; note that  $\text{dom}(f_M) = \text{dom}(\text{acc}_M) = L(M)$ , the language accepted by  $M$ .

Abusing ordinary function notation, for two partial, multi-valued functions  $f, g$ , we may write, e.g.,  $f(x, g(x))$  to indicate the partial, multivalued function  $c(x)$  with  $c(x) \mapsto z$  iff there is an output  $g(x) \mapsto y$  such that  $f(x, y) \mapsto z$ . (This also corresponds to composition of relations.)

We refer to a nondeterministic, polynomial-time machine  $M$  as “an NP machine”.

NPMV is the class of (possibly) partial, multi-valued functions computed by NP machines. NPSV is the subclass of NPMV consisting of those functions that are in fact single-valued, though they may still be partial and may have multiple accepting branches (as long as they all make the same output). PF is the class of functions computed by NP machines that use no nondeterminism, but may still be partial (viz., if they reject some input). FP is the class of total functions computed by deterministic polynomial-time machines.

A multivalued function  $f$  is a *refinement* of a multivalued function  $g$  if  $\text{dom}(f) = \text{dom}(g)$  and  $\text{set-}f(x) \subseteq \text{set-}g(x)$  for all  $x$  (this second condition is equivalent to  $f(x) \mapsto y$  implying  $g(x) \mapsto y$ ). If  $\mathcal{F}_1, \mathcal{F}_2$  are two classes of functions, we write  $\mathcal{F}_1 \subseteq_c \mathcal{F}_2$  if every  $f \in \mathcal{F}_1$  has a refinement in  $\mathcal{F}_2$ .

For any function  $f$ —whether it be injective or not, single-valued or not, total or not, surjective or not—we define  $f^{-1}$  to be the partial multivalued function defined by  $f^{-1}(y) \mapsto x$  iff  $f(x) \mapsto y$ . Given a class  $\mathcal{F}$  of functions, we say  $f$  is invertible in  $\mathcal{F}$  if  $f^{-1}$  has a refinement in  $\mathcal{F}$ . By our symbolic conventions, note that even though  $f^{-1}$  may be partial and multivalued, we always have  $f^{-1}(f(x)) \mapsto x$ . We also have  $f(f^{-1}(x)) \mapsto x$  iff  $x$  is in the image of  $f$ .

A function  $f$  is (polynomially) *honest* if there is a polynomial  $p$  such that for every input  $x$ , there is some output  $y$  such that  $|x| \leq p(|y|)$ . Note that honesty is a necessary condition to have some refinement of  $f^{-1}$  be computable in polynomial space, let alone polynomial time.

For these function classes, there are a number of subscripts to use that denote subclasses:

- The subscript  $t$  denotes the subclass of total functions, that is, whose domain is all of  $\Sigma^*$
- The subscript  $h$  denotes the subclass of honest functions
- The subscript  $g$  denotes the subclass of functions whose *graph*,  $\{(x, y) : f(x) \mapsto y\}$ , is decidable in polynomial time.

These subscripts may be used in any combination, e. g.  $\text{NPMV}_{ght}$ .

Selman [63] and Hemaspaandra, Naik, Ogihara, and Selman [35] showed that

$$\text{NPSV}_t = \text{FP}^{\text{NP} \cap \text{coNP}}.$$

**Theorem/Definition Q** ([14, Theorem 2]) Hypothesis Q is any of the following equivalent statements

- Q1 For all NP machines  $M$  that accept  $\Sigma^*$ , there exists a polynomial-time computable function  $g_M$  such that for all  $x$ ,  $g_M(x)$  outputs an accepting computation of  $M$  on  $x$ .
- Q2 All polynomial-time computable onto honest functions are invertible in PF.
- Q3  $\text{NPMV}_t \subseteq_c \text{FP}$ .
- Q4 For all  $S \in \text{P}$  such that  $S \subseteq \text{SAT}$ , there exists a polynomial-time computable  $g$  such that for all  $x \in S$ ,  $g(x)$  outputs a satisfying assignment of  $x$ .
- Q5  $\text{P} = \text{NP} \cap \text{coNP}$  and  $\text{NPMV}_t \subseteq_c \text{NPSV}_t$ .
- Q6 For all NP machines  $M$  such that  $L(M) = \text{SAT}$ , there is  $f_M \in \text{PF}$  such that for all  $\varphi \in \text{SAT}$ ,  $\varphi(f_M(\varphi, \text{acc}_M(\varphi))) = 1$ , that is,  $f_M$  takes in  $\varphi$  and any accepting computation of  $M$  on  $\varphi$ , and outputs at least one satisfying assignment to  $\varphi$ , and all such outputs satisfy  $\varphi$ .
- Q7 For all NP machines  $M, N$  such that  $L(M) \subseteq L(N)$ , there is  $f_M \in \text{PF}$  such that for all  $x \in L(M)$ ,  $f_M(x, \text{acc}_M(x))$  makes some output, and all its outputs are accepting computations of  $N(x)$ .
- Q8 For all  $L \in \text{P}$  and all NP machines  $M$  accepting  $L$ , there is  $f_M \in \text{PF}$  such that  $f_M(x)$  is an accepting computation of  $M(x)$ .

### 3 Polynomial-time Axioms of Choice

The formulations in this section are mostly based on choice functions for collections of sets. If  $S = \{S_i : i \in I\}$  is a collection of sets (where  $I$  is an arbitrary index set), then a choice function for  $S$  is a function  $f$  such that  $f(i) \in S_i$  for all  $i \in I$ .

**Definition 3.1** (Collection of languages, honestly non-empty, choice function) Given a complexity class  $\mathcal{C}$ , a  $\mathcal{C}$  collection of languages is a single language  $L \in \mathcal{C}$  such that each  $L_x := \{y : (x, y) \in L\}$  is also in  $\mathcal{C}$ .

A collection  $L$  of languages is *honestly non-empty* if there are polynomials  $p, q$  of positive degree such that, for every  $x$ ,  $L_x$  contains at least one string  $y$  of length  $p(|x|) \leq |y| \leq q(|x|)$ . We call  $p, q$  “honesty (lower, resp. upper) bounds” for  $L$ .

A partial, multi-valued function  $f$  is a *choice function* for a collection  $L$  of languages if: (1) for all  $x$ ,  $\text{set-}f(x) \subseteq L_x$ , and (2) for every nonempty  $L_x$ ,  $\text{set-}f(x)$  is nonempty.

*Remark 3.2* The polynomial upper bound in the definition of honestly non-empty is easy enough to justify at this point: there should at least exist a  $y$  in  $L_x$  that did not require super-polynomial-space to write down (so, e.g., if  $L \in \text{PSPACE}$ , then such a  $y$  could be found by a polynomial-space machine). But the polynomial lower bound may seem less intuitive. It turned out to be a very natural and useful condition at several points, e.g. Observation 3.6 and Proposition 3.8(1).

**Classical AC 1** (See [53, AC1, p. 5]). Every collection of non-empty sets has a choice function.

**Polynomial-Time AC (PAC) 1** Every  $\text{P}$  collection of honestly non-empty languages has a choice function in  $\text{FP}$ .

**Proposition 3.3** *PAC1 holds iff  $\text{NPMV}_{gt} \subseteq_c \text{FP}$ .*

*Proof* ( $\Rightarrow$ ) Suppose PAC1 holds. Let  $f$  be an  $\text{NPMV}_{gt}$  function and let  $L$  be the graph of  $f$ . By definition of  $\text{NPMV}_g$ , the graph of  $f$  is in  $\text{P}$ , so  $L \in \text{P}$ . Since  $f$  is total, every  $L_x$  is non-empty. Since each branch of  $f$  is polynomial-time, every output  $y$  of  $f(x)$  has  $|y| \leq \text{poly}(|x|)$ . To get a polynomial lower bound as well, we modify  $f$  to  $\hat{f}$ , defined by  $\hat{f}(x) \mapsto (x, y) \Leftrightarrow f(x) \mapsto y$ , so that every output now has  $|x| \leq |\hat{f}(x)|$ . Let  $\hat{L}$  be the graph of  $\hat{f}$ ;  $\hat{L}$  is now a  $\text{P}$  collection of honestly non-empty languages. Thus, by assumption, there is a polynomial-time choice function  $\hat{g}$  for  $\hat{L}$ . By construction of  $\hat{L}$ ,  $\hat{g}(x)$  always has the form  $(x, y)$ . Finally, define  $g(x) = y \Leftrightarrow \hat{g}(x) = (x, y)$ . It is clear that  $g$  is also in  $\text{FP}$  and refines  $f$ .

( $\Leftarrow$ ) Conversely, suppose  $\text{NPMV}_{gt} \subseteq_c \text{FP}$ , and let  $L$  be a  $\text{P}$  collection of honestly non-empty languages, with honesty bounds  $p, q$ . Define  $f$  by

$$\text{set-}f(x) = \{y : p(|x|) \leq |y| \leq q(|x|) \text{ and } (x, y) \in L\}$$

Since  $L \in \text{P}$ , the graph of  $f$  is in  $\text{P}$ : use the polynomial-time decider for  $L$ , together with verifying the length relationship between  $x, y$ . Moreover, since  $L_x$  is honestly non-empty for every  $x$ ,  $f$  is total, so  $f$  is in  $\text{NPMV}_{gt}$ . By assumption  $f$  has an  $\text{FP}$  refinement, which is clearly a polynomial-time choice function for  $L$ .  $\square$

The same argument can be trivially modified to several other standard complexity hypotheses, by varying the constraints in PAC1. We give two such here:

**Proposition 3.4** *Every NP collection of honestly non-empty languages has...*

1. ...an  $\text{NPSV}_t$  choice function iff  $\text{NPMV}_t \subseteq_c \text{NPSV}_t$ .
2. ...an  $\text{FP}$  choice function iff  $\text{NPMV}_t \subseteq_c \text{FP}$ ; note the latter is Hypothesis Q3.

Thus we see that Hypothesis Q is equivalent to a mixed nondeterministic/deterministic polynomial-time version of the Axiom of Choice. We will see below that Polynomial-Time AC 7 is also equivalent to Q.

*Proof* Let  $\mathcal{F} \in \{\text{NPSV}_t, \text{FP}\}$ .



( $\Rightarrow$ ) Suppose every NP collection of honestly non-empty languages has an  $\mathcal{F}$  choice function. Let  $f$  be in  $\text{NPMV}_t$ . As before, let  $\hat{f}(x) \mapsto (x, y) \Leftrightarrow f(x) \mapsto y$ , and let  $\hat{L}$  be the graph of  $\hat{f}$ . Since every NPMV function has its graph in NP,  $\hat{L}$  is in NP. As above,  $\hat{L}$  is an honestly non-empty collection of languages. By assumption, there is a choice function  $\hat{g} \in \mathcal{F}$  for  $\hat{L}$ . Define  $g(x)$  by  $g(x) = y \Leftrightarrow \hat{g}(x) = (x, y)$ ; for either choice of  $\mathcal{F}$  it is clear that  $\hat{g} \in \mathcal{F} \Leftrightarrow g \in \mathcal{F}$ . As above,  $g$  is clearly a refinement of  $f$ , and thus  $\text{NPMV}_t \subseteq_c \mathcal{F}$ .

( $\Leftarrow$ ) Conversely, suppose  $\text{NPMV}_t \subseteq_c \mathcal{F}$ , and let  $L$  be an NP collection of honestly non-empty languages, with honesty bounds  $p, q$ . Define  $f$  by

$$\text{set-}f(x) = \{y : p(|x|) \leq |y| \leq q(|x|) \text{ and } (x, y) \in L\}.$$

Since  $L \in \text{NP}$ , the graph of  $f$  is in NP: use the polynomial-time verifier for  $L$ , together with verifying the length relationship between  $x, y$ . Moreover, since  $L_x$  is honestly non-empty for every  $x$ ,  $f$  is total, so  $f$  is in  $\text{NPMV}_t$ . By assumption  $f$  has an  $\mathcal{F}$  refinement, which is clearly an  $\mathcal{F}$  choice function for  $L$ .  $\square$

**Classical AC 2** (See [53, AC2, p. 5], [37, Proposition 2.1]) Given any set  $X$  of pairwise disjoint nonempty sets, there is a set  $C$  that contains exactly one element from each set in  $X$ .

**Definition 3.5** (Transversal) Given a collection  $L$  of languages, a transversal for  $L$  is a language  $C$  such that  $C \cap L_x$  has exactly one element, for each  $x$  such that  $L_x$  is nonempty.

**Polynomial-Time AC (PAC) 2** Every polynomial-time collection of honestly nonempty languages that are pairwise disjoint, has a transversal in P.

While we have not been able to get a polynomial-time upper bound on transversals of such collections, we can give an upper bound within the second level of PH. More specifically, recall that  $\text{DP} = \{L \subseteq \Sigma^* : L = L_1 \setminus L_2 \text{ for some } L_1, L_2 \in \text{NP}\}$ . DP is contained in  $\text{P}^{\text{NP}}$ , and even in  $\text{P}^{\text{NP}[2]}$  (the machine only makes 2 queries on inputs of length  $n$ ). For any language  $L$ , we relativize DP to  $\text{DP}^L$ , by taking the preceding definition and replacing NP with  $\text{NP}^L$ .

**Observation 3.6** Every collection  $L$  of pairwise disjoint, honestly non-empty languages has a transversal in  $\text{DP}^L$ .

*Proof* Let  $p, q$  be honesty bounds for  $L$ , so that each  $L_x$  contains some  $y$  with  $p(|x|) \leq |y| \leq q(|x|)$ . Let  $C$  consist of the lexicographically first element of  $L_x$  with length at least  $p(|x|)$ , for each  $x \in \Sigma^*$ . Since the  $L_x$  are pairwise disjoint,  $C$  does not contain more than one element of each  $L_x$ , and thus it contains exactly one and is a transversal.

We now show that membership in  $C$  can be decided in  $\text{DP}^L$ . Let  $L_1 = \{y : (\exists x)[q^{-1}(|y|) \leq |x| \leq p^{-1}(|y|) \text{ and } (x, y) \in L]\}$  and let  $L_2 = \{y : (\exists y' <_{\text{lex}} y)(\exists x)[\max\{q^{-1}(|y'|), q^{-1}(|y|\}) \leq |x| \leq \min\{p^{-1}(|y|), p^{-1}(|y'|\}) \text{ and } (x, y) \in L \text{ and } (x, y') \in L]\}$ . Note that  $L_1, L_2$  are both in  $\text{NP}^L$ , and we have  $C = L_1 \setminus L_2$ .  $\square$

Thus for collections in P, there always exists an DP transversal.

**Open Question 3.7** *Is there a P collection of pairwise disjoint, honestly non-empty polynomial-time languages such that every transversal for L is NP-hard? NP-complete? DP-complete?*

While we do not have a clean equivalence between PAC2 and a more standard complexity hypothesis, we do have such a clean equivalence between more standard hypotheses and some NP versions of PAC2.

**Proposition 3.8** 1.  $\text{NPMV}_{gt} \subseteq_c \text{NPSV}_t$  iff

(\*) every P collection of honestly nonempty, pairwise disjoint languages has a transversal in NP.

2.  $\text{NPMV}_t \subseteq_c \text{NPSV}_t$  iff

(\*) every NP collection of honestly nonempty, pairwise disjoint languages has a transversal in NP.

*Proof* (1) Suppose (\*) holds, and let  $f \in \text{NPMV}_{gt}$ . Let  $L$  be  $\{(x, (x, y)) : x \in \Sigma^*, f(x) \mapsto y\}$  be a modified version of the graph of  $f$ . By definition the graph of  $f$  is in P, so  $L$  is also in P. Since  $f$  is total and each nondeterministic branch uses at most polynomial time, every  $L_x$  is honestly nonempty. Since every element of  $L_x$  is of the form  $(x, *)$ ,  $L_x$  and  $L_y$  are disjoint when  $x \neq y$ . Hence there is a set  $C \in \text{NP}$  that contains exactly one element from each  $L_x$ . We define an  $\text{NPSV}_{gt}$  function  $g$  that, on input  $x$ , nondeterministically guesses  $y$  and  $w$ , uses  $w$  as a witness to verify that  $(x, y) \in C$ , and outputs  $y$ . This  $g$  is an  $\text{NPSV}_t$  refinement of  $f$ .

Conversely, suppose  $\text{NPMV}_{gt} \subseteq_c \text{NPSV}_t$ , and let  $L$  be a P collection of honestly nonempty, pairwise disjoint languages with honesty bounds  $p, q$ . Define  $f$  by

$$\text{set-}f(x) = \{y : p(|x|) \leq |y| \leq q(|x|) \text{ and } (x, y) \in L\}.$$

Since  $L \in \text{P}$ , the graph of  $f$  is in P. Moreover, since  $L_x$  is honestly non-empty for every  $x$ ,  $f$  is total. Hence  $f$  has an  $\text{NPSV}_t$  refinement  $g$ . Let  $C$  be the image of  $g$ . It is clear that  $C$  intersects each  $L_x$  exactly once. The following nondeterministic machine shows that  $C$  is in NP: on input  $y$ , it guesses  $x$  such that  $p(|x|) \leq |y|$  (so in particular  $|x| \leq \text{poly}(|y|)$ ), guesses an accepting path of  $g$ , simulates the execution of  $g(x)$  on that accepting path, and accepts if the output on that path is equal to  $y$ .

(2) We observe here how the preceding parts of the proof change for the second part. For  $f \in \text{NPMV}_t$ , its graph (and modified graph as above) will be in NP. (\*) then allows us to conclude existence of a  $C \in \text{NP}$  as before, and the rest of the argument goes through. For the other direction of the argument, if  $L$  is only in NP, then  $f$  will still be in  $\text{NPMV}_t$ , so the assumption  $\text{NPMV}_t \subseteq_c \text{NPSV}_t$  is still enough to get the  $\text{NPSV}_t$  refinement, and again the rest of the proof goes through.  $\square$

We note that it is only in the final step of the proof—in which a nondeterministic machine must guess a preimage  $x \in g^{-1}(y)$ —that nondeterminism seems to be needed, which is the only obstacle standing in the way of an equivalence with PAC2.

The preceding results let us show that two NP versions of AC are equivalent:

**Corollary 3.9** (“NP Axiom of Choice 1=NP Axiom of Choice 2”) *Every NP collection of honestly non-empty languages has an NPSV choice function iff every NP collection of pairwise disjoint honestly non-empty languages has an NP transversal.*

*Proof* Proposition 3.4(1) says that the first condition in the statement here is equivalent to  $\text{NPMV}_t \subseteq_c \text{NPSV}_t$ , and Proposition 3.8(2) shows the latter is equivalent to the second condition in the statement here.  $\square$

**Observation 3.10** *PAC2 implies condition (\*) from Proposition 3.8.*

*Proof* The conclusion of PAC2 implies existence of a P transversal, while the conclusion of (\*) merely requires an NP transversal, but otherwise the two are the same.  $\square$

**Classical AC 3** (See [53, AC5, p. 5]) For every function  $f$  there is a function  $g$  such that  $\text{dom}(g) = \text{img}(f)$  and for every  $x \in \text{dom}(g)$ ,  $f(g(x)) = x$ .

**Polynomial-Time AC (PAC) 3** For every partial polynomial-time honest function  $f \in \text{PF}$ , there is a  $g \in \text{PF}$  with  $\text{dom}(g) = \text{img}(f)$  such that for every  $x \in \text{dom}(g)$ ,  $f(g(x)) = x$ .

Recall that a (worst-case) *one-way function* is a one-one, honest function  $f \in \text{PF}$  such that  $f^{-1} \notin \text{PF}$ , so PAC3 is simply the statement that one-way functions do not exist. This was one of the topics Selman worked on:

**Theorem 3.11** (Grollman and Selman [30] and Ko [39]) *PAC3 holds iff  $\text{P} = \text{UP}$ .*

## 4 Polynomial-Time Cardinality

### 4.1 Definition and Basic Properties

Several formulations of the Axiom of Choice are based on cardinalities. Here we develop polynomial-time analogues of these. This development is also partially motivated by the quotation from Downey [12] in Section 1.

Let  $\Sigma, \Gamma$  be two finite alphabets. For  $A \subseteq \Sigma^*$  and  $B \subseteq \Gamma^*$ , by a “partial function  $f: A \rightarrow B$ ” we mean a partial function  $f: \Sigma^* \rightarrow \Gamma^*$  such that  $A \subseteq \text{dom}(f)$  and  $f(A) \subseteq B$ . For inputs  $x \notin A$ , we impose no restrictions on  $f(x)$ —it might be in  $B$ , outside of  $B$ , or undefined. When we speak of properties of such a partial function—e. g., being computable in polynomial time, being injective, surjective, bijective, etc.—we only refer to these properties as they apply to inputs in  $A$  and outputs in  $B$ . For example, a partial function  $f: A \rightarrow B$  is injective iff for all distinct  $x, y \in A$ ,  $f(x)$  and  $f(y)$  are distinct. Even if  $f$  is defined on inputs outside of  $A$ , it may be non-injective on those inputs, or even have  $f(x) = f(a)$  for some  $x \notin A, a \in A$ , and still be considered an injective partial function  $A \rightarrow B$ . We say a partial function  $f: A \rightarrow B$  is polynomial-time invertible, or p-invertible, if  $f^{-1}: f(A) \rightarrow A$  is single-valued (equivalently,  $f$  is injective) and polynomial-time

computable on  $f(A)$ . A partial function  $f$  is *length-increasing* if  $|f(x)| > |x|$  for all  $x \in \text{dom}(f)$ .

**Definition 4.1** Two sets  $A \subseteq \Sigma^*$ ,  $B \subseteq \Gamma^*$  have the same *p-cardinality* or are *p-equipollent* if there are partial polynomial-time computable functions  $f: A \rightarrow B$  and  $g: B \rightarrow A$  such that  $f \circ g = \text{id}_B$  and  $g \circ f = \text{id}_A$ . In this case we call  $f$  a *p-equipollence*  $A \rightarrow B$ .

We denote the p-cardinality of  $A$ —or, the same, its p-equipollence class—by  $\|A\|_p$ , so  $\|A\|_p = \|B\|_p$  means that  $A$  and  $B$  are p-equipollent.

We say that the p-cardinality of  $A$  is at most that of  $B$ , denoted  $\|A\|_p \leq \|B\|_p$ , if  $\|A\|_p = \|B'\|_p$  for some subset  $B' \subseteq B$ .  $\|A\|_p < \|B\|_p$  denotes  $\|A\|_p \leq \|B\|_p$  and  $\|A\|_p \neq \|B\|_p$ .

The relation  $\leq$  between p-cardinals is transitive and reflexive, but we will see in Proposition 4.4 that it is *not* a partial order, that is, there exist  $A, B$  for which  $\|A\|_p \leq \|B\|_p$  and  $\|B\|_p \leq \|A\|_p$  but  $\|A\|_p \neq \|B\|_p$ ; however see Theorem 4.10 for a partial positive result. This is why we maintain the notation  $\leq$  (rather than, say,  $\preceq$ , as is typically done for cardinals). In Proposition 4.39 we will also see that it is not total (there are incomparable p-cardinals).

As we do with many-one reductions, it would make sense to introduce an equivalence relation  $\|A\|_p \equiv \|B\|_p$  defined by  $\|A\|_p \leq \|B\|_p$  and  $\|B\|_p \leq \|A\|_p$ ; roughly speaking, the relationship between equivalence and equality of p-cardinals is analogous to the relationship between Karp equivalence and p-isomorphism of languages. The quotient relation  $\leq$  on equivalence classes of p-cardinals is a partial order by construction, but Proposition 4.39 still shows it to be only partial, not total. We will not use  $\equiv$  much, though there is surely interesting theory to be explored there.

*Notational note.* We have tried to consistently use the superscript<sup>*p*</sup> to denote a polynomial-time version of something, but in the case of cardinality, we wanted to use  $\|\cdot\|$  as it is a standard symbol for cardinality, but wanted to avoid  $\|\cdot\|^p$  because of its possible confusion with raising to the *p*-th power.

**Proposition 4.2** For languages  $A \subseteq \Sigma^*$ ,  $B \subseteq \Gamma^*$ , if  $\|A\|_p = \|B\|_p$ , then  $A \equiv_m^p B$  unless  $A = \Sigma^*$  or  $B = \Gamma^*$ . In the latter case, both of  $A$  and  $B$  are in  $\mathcal{P}$ .

*Proof* Suppose that  $\|A\|_p = \|B\|_p$ , and first also suppose that  $A \neq \Sigma^*$  and  $B \neq \Gamma^*$ . We will show that  $B \leq_m^p A$ ; the reverse follows by symmetry. Let  $a_0 \in \bar{A}$ .

Let  $f: A \rightarrow B$  be a p-equipollence. Let  $p$  be a polynomial and  $M, N$  Turing machine transducers such that  $M$  computes  $f$  on inputs in  $A$  in polynomial time and  $N$  computes  $f^{-1}$  on inputs in  $B$  in time  $\leq p(n)$ . The following function  $r$  is a polynomial-time reduction from  $B$  to  $A$ : on input  $x$ , run  $N(x)$  for  $p(|x|)$  steps. If  $N$  rejects or makes no output by that time, then  $r(x) := a_0$ . For  $B \subseteq \text{dom}(f^{-1}) \subseteq \text{dom}(N)$ . If  $N(x) = y$  by the time it has made  $p(|x|)$  steps, check that  $M(y) = x$ . If not, then output  $a_0$  (for if  $x$  were in  $B$  then we have  $M(N(x)) = f(f^{-1}(x)) = x$ ). If so, then let  $r(x) = y$ , for at this point  $x \in B$  iff  $r(x) \in A$ . (Note that it is possible that  $\text{dom}(N)$  is strictly larger than  $B$  and that  $\text{dom}(M)$  is strictly larger than  $A$ , and that  $M(N(x)) = x$  for some  $x$  outside  $B$ , so the final check that  $r(x) \in A$  is needed.)

If  $A = \Sigma^*$ , then there is no  $a_0$  to use, but in those cases the above reduction can directly decide whether the input  $x$  was in  $B$ , so we get  $B \in \mathcal{P}$ .  $\square$

Note that in the above proof we really used the fact that we had access to (partial) polynomial-time computable functions that were inverses of one another. If one tried to extend Nerode & Remmel’s theory [50, 51] from unary languages to larger alphabets using only polynomial-time, bijective, honest functions, but without the stipulation of a polynomial-time inverse (even if one required such functions in both directions, but without requiring them to be inverses of one another) such a result seems much less likely to hold.

We note that the preceding result does *not* extend in the natural way to  $\|A\|_p \leq \|B\|_p$ :

**Proposition 4.3** *There exist infinite, co-infinite sets  $A, B$  such that  $\|A\|_p \leq \|B\|_p$  but  $A \not\leq_T^p B$ .*

*Proof* Let  $A$  be an EXP-complete set and  $B = 0\Sigma^*$  (the prefix 0 is just to ensure that  $B$  is co-infinite). Then  $0A = \{0a : a \in A\}$  is EXP-complete, and  $0A \subseteq B$ , so  $\|A\|_p = \|0A\|_p \leq \|B\|_p$ . But by the Time Hierarchy Theorem,  $A \not\leq_T^p B$ , for otherwise we would have  $A \in \mathcal{P}$ .  $\square$

We can use Proposition 4.2 to show that  $\leq$  is *not* a partial order:

**Proposition 4.4** ( $\leq$  is not a partial order on p-cardinals) *There exist languages  $A, B \subseteq \Sigma^*$  such that  $\|A\|_p \leq \|B\|_p$  and  $\|B\|_p \leq \|A\|_p$  (we might denote these together as  $\|A\|_p \equiv \|B\|_p$ ), but  $\|A\|_p \neq \|B\|_p$ .*

*Proof* Let  $A = 1\Sigma^*$ ,  $B = 1(\Sigma^* \oplus B')$  for some  $B' \notin \mathcal{P}$ . First, we have  $\|A\|_p = \|10\Sigma^*\|_p$ , and  $10\Sigma^* \subseteq 1(\Sigma^* \oplus B')$  (for any  $B'$ ), so we get  $\|A\|_p \leq \|B\|_p$ . In the opposite direction, since  $B \subseteq A$ , we get  $\|B\|_p \leq \|A\|_p$ . But if  $\|A\|_p = \|B\|_p$ , Proposition 4.2 would then give  $A \equiv_m^p B$  (since neither is  $\Sigma^*$ ), implying that  $B \in \mathcal{P}$ , contradicting the fact that  $B \equiv_m^p B' \notin \mathcal{P}$ .  $\square$

### 4.2 Comparison with p-isomorphism

Note that, a priori, having the same p-cardinality is a weaker condition than being p-isomorphic, since the bijections required for p-cardinality need not be total, nor be bijections on all of  $\Sigma^*$ . Our first two results in this direction help clarify the relationship between the two.

**Proposition 4.5** *If  $A \cong^p B$ , then  $\|A\|_p = \|B\|_p$  and  $\|\bar{A}\|_p = \|\bar{B}\|_p$ . The converse holds if  $A$  is in  $\mathcal{P}$ .*

After the fact, we discovered that our argument for the converse here is very similar to, but more general than, one from Goldsmith, Hemachandra, and Kunen [27], reproduced below as Corollary 4.19(2).

*Proof* Suppose  $f : \Sigma^* \rightarrow \Gamma^*$  is a p-isomorphism  $A \rightarrow B$ . Then  $f|_A : A \rightarrow B$  is a p-equipollence, and similarly  $f|_{\bar{A}} : \bar{A} \rightarrow \bar{B}$  is a p-equipollence.

Suppose conversely that  $f: A \rightarrow B$  is a  $p$ -equipollence (on  $A$ ) and  $g: \bar{A} \rightarrow \bar{B}$  is a  $p$ -equipollence, and furthermore that  $A \in \mathbf{P}$ . Then the following is a  $p$ -bijection  $\phi: \Sigma^* \rightarrow \Gamma^*$  that is a  $p$ -isomorphism witnessing  $A \cong^p B$ . On input  $x$ , first we use the poly-time machine to decide whether  $x \in A$ . If so, then output  $f(x)$ , and if not, then output  $g(x)$ . Since  $\Sigma^*$  is the disjoint union of  $A$  and  $\bar{A}$ ,  $\phi$  is a total polynomial-time bijection.

The following algorithm computes  $\phi^{-1}$  in polynomial time. Let  $t(n)$  be a polynomial upper bound on the running time of some algorithm computing  $f^{-1}|_B$  and also of some algorithm computing  $g^{-1}|_{\bar{B}}$ . On input  $x \in \Gamma^*$ , try computing both  $f^{-1}(x)$  and  $g^{-1}(x)$  for  $t(n)$  steps. If only one of them finishes in that amount of time, then we know whether  $x$  was in  $B$  or  $\bar{B}$ , and we use that as our output. However, note that the algorithm computing  $f^{-1}$  may have domain larger than  $B$ , and the algorithm computing  $g^{-1}$  may have domain larger than  $\bar{B}$ , so it is possible that both  $f^{-1}(x)$  and  $g^{-1}(x)$  output strings  $y, z$ , respectively, within  $t(|x|)$  steps. If  $y = z$  then the algorithm may unambiguously output  $y$  without having determined whether  $x \in B$  or not. If  $y \neq z$ , then we compute  $\phi(y)$  and  $\phi(z)$ . Since  $\phi$  is a bijection and  $y \neq z$ , we have  $\phi(y) \neq \phi(z)$ , so only one of them can be equal to  $x$ . (And at least one of them must equal  $x$ , since  $x$  must be in at least one of  $B, \bar{B}$ .) If  $\phi(y) = x$ , then output  $y$ ; if  $\phi(z) = x$ , then output  $z$ .  $\square$

- Proposition 4.6** 1. *There is a  $p$ -cardinal containing infinitely many distinct  $p$ -isomorphism classes.*  
 2. *There exist non- $p$ -isomorphic sets  $A, B$  of the same  $p$ -cardinality in which  $\bar{A}, \bar{B}$  are both infinite and in EXP.*

Note that there are no such examples where  $A, B$  are finite, for in that case  $\|A\|_p = \|B\|_p$  implies  $A \cong^p B$ .

*Proof* 1. For each  $n \in \mathbb{N}$ , let  $S_n$  be the set of the lexicographically first  $n$  strings. Define  $A_n := \Sigma^* \setminus S_n$ . Then all  $A_n$  are  $p$ -equipollent to  $A_0 = \Sigma^*$ : the  $p$ -equipollence  $A_0 \rightarrow A_n$  simply shifts every string “up  $n$ ” in terms of the corresponding  $|\Sigma|$ -ary numbers. But if  $n \neq n'$ , then  $\bar{A}_n$  and  $\bar{A}_{n'}$  are finite sets of different sizes, so they are not  $p$ -equipollent. Since  $\|\bar{A}_n\|_p \neq \|\bar{A}_{n'}\|_p$  in this case, they cannot be  $p$ -isomorphic, hence neither can  $A_n$  and  $A_{n'}$ .

2. We give an example with  $\bar{A}, \bar{B}$  both infinite and in EXP. Recall that a set  $A$  is  $\mathbf{P}$ -immune if it is infinite but contains no infinite subset in  $\mathbf{P}$ . Let  $A$  be any co- $\mathbf{P}$ -immune set, that is,  $\bar{A}$  is infinite but contains no infinite subsets in  $\mathbf{P}$ . For example, Ko & Moore [40] showed that  $\mathbf{P}$ -bi-immune sets, in which both  $A$  and its complement are  $\mathbf{P}$ -immune, exist in EXP. Let  $B = \{1x : x \in A\}$ . Then a  $p$ -equipollence is given by  $A \ni a \mapsto 1a \in B$ , with inverse  $B \ni 1x \mapsto x \in A$ . But we claim that  $A \not\cong^p B$ . The short version is:  $\mathbf{P}$ -immunity is preserved by isomorphism, but  $\bar{B}$  contains  $0\Sigma^*$  as an infinite subset in  $\mathbf{P}$ . In more detail: suppose there is a polynomial-time computable bijection  $f: \Sigma^* \rightarrow \Gamma^*$  such that  $f(A) = B$  and with  $f^{-1} \in \mathbf{FP}$  as well. Since  $f^{-1}(\bar{B}) = \bar{A}$ , we have that  $f^{-1}(0\Sigma^*) \subseteq \bar{A}$ . But  $f^{-1}(0\Sigma^*)$  is in  $\mathbf{P}$ , for  $x \in f^{-1}(0\Sigma^*)$  iff  $f(x)$  begins with a 0, which is easily checked. As  $0\Sigma^*$  is infinite and  $f^{-1}$  is a bijection, this contradicts the  $\mathbf{P}$ -immunity of  $\bar{A}$ .  $\square$

**Observation 4.7** Let  $L \notin \mathcal{P}$ . If  $L \equiv_m^p L'$  implies  $L \cong^p L'$ , then the  $p$ -isomorphism class of  $L$  is equal to its  $p$ -equipollence class.

*Proof* Since  $L \notin \mathcal{P}$ ,  $\Sigma^* \not\equiv_m^p L$ , so the  $p$ -isomorphism class of  $L$  does not include  $\Sigma^*$ . Since  $p$ -isomorphism implies  $p$ -equipollence, and  $p$ -equipollence (avoiding  $\Sigma^*$ ) implies  $\equiv_m^p$  (Proposition 4.2), the result follows.  $\square$

**Corollary 4.8** (cf. Kurtz, Mahaney, Royer [41]) For every set  $A$ , there is a set  $B \geq_m^p A$  such that the  $p$ -isomorphism class of  $B$  is equal to its  $p$ -equipollence class. There exists a set that is  $\leq_{bit}^p$ -complete for EXP whose  $p$ -isomorphism class is equal to its  $p$ -equipollence class.

*Proof* Kurtz, Mahaney, and Royer [41] showed for every  $A$  there is a set  $B \geq_m^p A$  such that the polynomial-time many-one degree of  $B$  was equal to its  $p$ -isomorphism class. Such  $B$  is necessarily not in  $\mathcal{P}$ , since  $\mathcal{P}$  is its own poly-time many-one degree, but contains pairs of non- $p$ -isomorphic sets. So we may apply Observation 4.7. They similarly showed a collapse result for a  $\leq_{bit}^p$ -complete set for EXP, which is not in  $\mathcal{P}$  by the Time Hierarchy Theorem, so again we may apply Observation 4.7.  $\square$

In Proposition 4.6 we exhibited a  $p$ -cardinal (that of  $\Sigma^*$ ) containing infinitely many  $p$ -isomorphism classes; we believe much more should be true, in fact, we expect such cardinals are dense in the  $\leq_m^p$  ordering:

**Conjecture 4.9** For every set  $A$ , there is a set  $B \geq_m^p A$  such that  $\|B\|_p$  contains infinitely many distinct  $p$ -isomorphism classes.

We next observe that Berman & Hartmanis's Cantor–Bernstein-like proof extends to  $p$ -cardinality:

**Theorem 4.10** Let  $A \subseteq \Sigma^*$ ,  $B \subseteq \Gamma^*$ , and let  $p: A \rightarrow B$  and  $q: B \rightarrow A$  be length-increasing,  $p$ -invertible partial polynomial-time injective functions. Then  $A$  and  $B$  have the same  $p$ -cardinality.

A proof nearly identical to that in [5, Theorem 1] works; for expository purposes, we follow the terminology and proof described in [15, Section 2].

*Proof* Define the following graph  $G_{p,q}$  on vertex set  $0A \cup 1B$ : for every  $x \in A$ , there is an edge  $0x \rightarrow 1p(x)$ , and for every  $y \in B$ , there is an edge  $1y \rightarrow 0q(y)$ . The graph is readily seen to be bipartite by construction. Since  $p$  and  $q$  are functions with  $\text{dom}(p) \supseteq A$  and  $\text{dom}(q) \supseteq B$ , every vertex has out-degree exactly 1. Since  $p$  and  $q$  are invertible, every vertex has in-degree at most 1. The connected components of  $G_{p,q}$  are called chains. We call a vertex a source if it has in-degree 0. Thus the graph is a disjoint union of chains, each of which is one of: finite cycles, two-way infinite paths, a one-way infinite path with its source in  $0A$ , or a one-way infinite path with its source in  $1B$ . Since  $p$  and  $q$  are length-increasing, there can be no finite cycles. There also cannot be any two-way infinite paths, since in following the chain

backwards, the length of the strings would have to be decreasing forever. Since every vertex has out-degree 1, every vertex is part of a chain.

Let  $C_A$  be the vertices that are part of a chain with source in  $0A$ , and  $C_B$  be the vertices in that are part of a chain with source in  $1B$ . We define

$$\phi(x) = \begin{cases} p(x) & \text{if } 0x \in 0A \cap C_A \\ q^{-1}(x) & \text{if } 0x \in 0A \cap C_B \end{cases} \quad \psi(y) = \begin{cases} p^{-1}(y) & \text{if } 1y \in 1B \cap C_A \\ q(y) & \text{if } 1y \in 1B \cap C_B \end{cases}$$

As is usual in such proofs, it is readily verified that  $\phi$  and  $\psi$  are inverses of one another.

Since  $p^{-1}$  need not have domain all of  $B$ , and  $q^{-1}$  need not have domain all of  $A$ , it remains to verify that  $\phi$  (resp.,  $\psi$ ) can be computed in PF on domain  $A$  (resp.,  $B$ ). We give the proof for  $\phi$ , the proof for  $\psi$  being similar. Let  $t(n)$  be a polynomial such that  $p^{-1}$  can be computed in  $t(n)$  time on inputs in  $A$  and  $q^{-1}$  can be computed in  $t(n)$  time on inputs in  $B$ . For inputs in  $A$ , we modify  $q^{-1}$  to a function  $\widehat{q^{-1}}$  such that  $\widehat{q^{-1}}(x) = q^{-1}(x)$  if  $x \in q(B)$ , and otherwise  $\widehat{q^{-1}}(x)$  outputs a special symbol  $\perp$ . The latter can be computed in  $O(t(n) \log t(n))$  time by running a  $t(n)$ -time machine for  $q^{-1}$ , with an additional clock, and if the clock ever hits  $t(n)$ , then stopping and outputting  $\perp$ . Similarly we modify  $p^{-1}$  to  $\widehat{p^{-1}}$ .

The key to computing  $\phi$  efficiently is to determine whether  $0x \in C_A$  or  $0x \in C_B$ . It does this by applying  $\widehat{q^{-1}}$  and  $\widehat{p^{-1}}$  alternately until it gets  $\perp$ , at which point it has found a source of the chain and thus knows which of  $C_A$  or  $C_B$  the vertex  $0x$  is in. Since  $p^{-1}$  and  $q^{-1}$  are length-decreasing, this takes no more than time polynomial in  $|x|$ .  $\square$

*Remark 4.11* We see from the proof that the main use of the length-increasing condition was that any sequence of  $p^{-1}, q^{-1}$  terminated after  $\text{poly}(|x|)$  steps. Thus the argument generalizes to any p-well-orderings (see Definition 6.1)  $\prec_A$  on  $A$  and  $\prec_B$  on  $B$  such that  $q(p(x)) \succ_A x$  and  $p(q(y)) \succ_B y$  for all  $x \in A, y \in B$ .

In fact, one can get away with further generalizations. If a sequence of  $p^{-1}, q^{-1}$  forms a cycle of p-bounded length, this can be detected and easily handled. The real issue with the above proof is cycles or chains such that a polynomial-time algorithm cannot identify when a string is part of such a cycle or chain. (A necessary condition is that such a cycle or chain is super-polynomially long, but there could still be very long such chains/cycles that are easily identifiable for other reasons.)

### 4.3 P-Countability: Density, Rankability, Enumerability, and Compressibility

It is natural to consider  $\Sigma^*$  to be the p-cardinality analogue of  $\aleph_0$ , the cardinality of  $\mathbb{N}$ , so here we examine sets that have the same p-cardinality as  $\Sigma^*$ .

**Definition 4.12** (p-countable) A set is *p-countable* if it is finite or has the same p-cardinality as  $\Sigma^*$ .

Before we get to interesting connections, we ensure that our exploration is not dependent on alphabet:



**Observation 4.13** Let  $\Sigma, \Gamma$  be two finite alphabets of size at least 2. Then  $\|\Sigma^*\|_p = \|\Gamma^*\|_p$ .

*Proof* Let  $k = |\Sigma|, \ell = |\Gamma|$ . The following is a  $p$ -equipollence  $\Sigma^* \rightarrow \Gamma^*$ . Given  $x \in \Sigma^*$ , consider the natural number  $N$  associated to it (see Section 2), and then let  $y$  be the  $\ell$ -ary string in  $\Gamma^*$  associated with  $N$ , and output  $y$ . We have that  $|x| \sim \log_k N$  and  $|y| \sim \log_\ell N$  (since both  $k, \ell \geq 2$ ), so  $|y| = \Theta(|x|)$ . It is clear that the computations can be done in polynomial time.  $\square$

Because every language is a subset of  $\Sigma^*$ , every  $A$  has  $\|A\|_p \leq \|\Sigma^*\|_p$ , so it would seem that  $\Sigma^*$  is the “largest”  $p$ -cardinal. But remember that  $\leq$  is only a pre-order, not a partial order, and in the proof of Proposition 4.4, we saw that for  $B' \notin \mathcal{P}$ ,  $\|\Sigma^*\|_p < \|\Sigma^* \oplus B'\|_p$  (even though we also have  $\|\Sigma^* \oplus B'\|_p \leq \|\Sigma^*\|_p$ ) but the two cardinalities are not equal. So the  $\equiv$ -equivalence class of  $\|\Sigma^*\|_p$  is indeed maximal in the quotient poset, but the  $p$ -cardinal itself is not maximal in the  $\leq$  preorder.

### 4.3.1 Enumerability

Selman introduced the notion of  $p$ -enumerability:

**Definition 4.14** ( $p$ -enumerable, Selman [56]) A language  $L$  is  $p$ -enumerable if there is a function  $f \in \text{FP}$  such that  $\text{img}(f) = L$  and such that for all  $y \in L$ , there exists  $x$  such that  $f(x) = y$  and  $|x| \leq \text{poly}(|y|)$ .

Note that in Selman’s definition,  $f$  need not be injective.

**Observation 4.15** A  $p$ -countable set is  $p$ -enumerable.

*Proof* For finite sets this is clear. For an infinite  $p$ -countable set  $L$ , let  $f: \Sigma^* \rightarrow L$  be a  $p$ -equipollence. Then  $L$  is  $p$ -enumerable via  $f$ ; the honesty-like condition is guaranteed by the  $p$ -computability of  $f^{-1}$ .  $\square$

This also follows from the fact that the  $p$ -enumerable sets are precisely those in NP [56, Corollary 2] and  $p$ -countable sets are in  $\mathcal{P}$  (by Proposition 4.2). This simple observation will have an interesting consequence when we get to arithmetic of  $p$ -cardinals (Proposition 4.36).

**Definition 4.16** ( $p$ -enumerable by iteration [32]) A language  $L$  is  $p$ -enumerable by iteration if there is a Turing machine  $M$  and polynomial  $p$  such that for all  $x \in L$ ,  $M(x)$  halts within  $p(|x|)$  steps, and there is some  $x_0 \in L$  such that  $L = \{x_0, M(x_0), M(M(x_0)), M(M(M(x_0))), \dots\}$ .  $L$  is invertibly  $p$ -enumerable by iteration if it is  $p$ -enumerable by iteration as before, and the function computed by  $M$  has a polynomial-time inverse on its image.

Hemachandra, Hoene, Siefkes, and Young [32, Section 5], among other things, gave examples of languages that were invertibly p-enumerate by iteration, including computably enumerable P-cylinders, and sets with various forms of self-reducibility. Here we add another family of examples:

**Proposition 4.17** *If  $L$  is p-countable then  $L$  is invertibly p-enumerable by iteration. The converse does not always hold.*

*Proof* For finite  $L$  this is clear, the enumerator implements a finite cycle through all the elements of  $L$ .

For infinite  $L$ , let  $f: L \rightarrow \Sigma^*$  be a p-equipollence. Let  $x_0 = f^{-1}(\epsilon)$ , and for  $x \in L$  define  $M(x)$  by  $M(x) = f^{-1}(f(x) + 1)$ . Since  $f, f^{-1}$  and addition by constants are all p-invertible, so is  $M$ .

Hemachandra et al. [32] give examples of languages invertibly p-enumerable by iteration that are not in P, but by Proposition 4.2, p-countable languages are in P.  $\square$

### 4.3.2 Compressibility

Goldsmith, Hemachandra, and Kunen [27] defined a language  $L$  to be P-compressible if there is an  $f \in \text{FP}$ ,  $f: \Sigma^* \rightarrow \Sigma^*$  such that  $f|_L: L \rightarrow \Sigma^*$  is a bijection.

**Observation 4.18** *If  $L$  is p-countably infinite, then  $L$  is p-compressible by a function that is p-invertible.*

*Proof* Given a p-equipollence  $f: L \rightarrow \Sigma^*$ , define  $\hat{f}: \Sigma^* \rightarrow \Sigma^*$  by

$$\hat{f}(x) = \begin{cases} x & x \notin L \\ f(x) & x \in L. \end{cases}$$

Since  $f \in \text{PF}$  with  $f|_L: L \rightarrow \Sigma^*$  a bijection, and  $L \in \text{P}$  (since it is p-countable), clearly  $\hat{f} \in \text{FP}$  and  $\hat{f}|_L: L \rightarrow \Sigma^*$  is still a bijection. Moreover,  $f^{-1}: \Sigma^* \rightarrow L$  is an inverse of  $\hat{f}$  that is in FP.  $\square$

Several results and techniques of Goldsmith, Hemachandra, and Kunen thus have immediate implications for p-countable sets; we record a particularly interesting one here.

**Corollary 4.19** (cf. Goldsmith, Hemachandra, Kunen [27, Proposition 3.18]) *If  $A, \bar{A}, B, \bar{B}$  are all p-countably infinite, then  $A \cong^p B$ .*

*Proof* Goldsmith–Hemachandra–Kunen [27, Proposition 3.18] use Grollman & Selman’s result that  $\text{P} = \text{UP}$  iff one-way functions do not exist [30]. Here instead, we use the fact that the p-equipollences with  $\Sigma^*$  have inverses by definition. But the following construction is otherwise the same as [27]. We include it here for completeness and ease of reference.

Let  $f_1 : A \rightarrow \Sigma^*$  and  $f_2 : \bar{A} \rightarrow \Sigma^*$  be p-equipollences, with extensions  $\hat{f}_1, \hat{f}_2$  to total FP functions. Define

$$f(x) = \begin{cases} 0\hat{f}_1(x) & x \in A \\ 1\hat{f}_2(x) & x \notin A. \end{cases}$$

Then we have

$$f^{-1}(by) = \begin{cases} f_1^{-1}(y) & b = 0 \\ f_2^{-1}(y) & b = 1. \end{cases}$$

Since  $\hat{f}_1, \hat{f}_2, f_1^{-1}, f_2^{-1}$ , and the characteristic function of  $A$  are all computable in polynomial time (the latter by Proposition 4.2), so are  $f$  and  $f^{-1}$ . Furthermore,  $f, f^{-1}$  are total bijections  $\Sigma^* \rightarrow \Sigma^*$ .

Define  $g$  similarly for  $B$ . Then  $h(x) = f^{-1}(g(x))$  is a total, p-equipollence  $\Sigma^* \rightarrow \Sigma^*$  with  $h(B) = f^{-1}(g(B)) = f^{-1}(0\hat{g}_1(B)) = f^{-1}(0g_1(B)) = f^{-1}(0\Sigma^*) = f_1^{-1}(\Sigma^*) = A$ , so  $h$  is the desired p-isomorphism  $B \rightarrow A$ . □

### 4.3.3 Rankability

Suppose  $L \subseteq \Sigma^*$  is p-countably infinite. Then there is a polynomial-time computable and invertible bijection  $f : L \rightarrow \Sigma^*$ . For infinite languages, ranking functions [1, 25] are a special case of bijective maps to  $\Sigma^*$ :

**Definition 4.20** (Ranking function, p-rankability [1, 25]) The (strong) ranking function of a language  $L$  is the map  $rk_L(x) = |\{y \in L : y \leq_{lex} x\}|$ . A language  $L$  is strongly p-rankable if its strong ranking function can be computed in FP, and weakly p-rankable if the ranking function can be computed in PF on those inputs that are actually in  $L$ .

From p-countability, we don't quite get p-rankability, but we get a version if we allow replacing the ordering  $\leq_{lex}$  by a different ordering that shares many properties with  $\leq_{lex}$ . We generalize rankability to arbitrary total orderings  $<$ . We define the strong ranking function of  $L$  with respect to  $<$  as  $rk_{L,<}(x) = |\{y \in L : y \leq x\}|$ , and we say  $L$  is strongly (resp., weakly) p-rankable with respect to  $<$  if  $rk_{L,<}$  is in FP (resp., in PF on inputs in  $L$ ).

**Definition 4.21** (Length-related [46]) A total ordering  $<$  on  $\Sigma^*$  is length-related if  $x < y$  implies  $|x| \leq \text{poly}(|y|)$ .

**Observation 4.22** If  $L$  is p-countable, then  $L$  is weakly p-rankable under some length-related total (on  $L$ ) ordering  $< \in \text{PF}$ .

*Proof* For finite sets this is clear. Let  $L$  be an infinite p-countable set, and let  $f : L \rightarrow \Sigma^*$  be a p-equipollence. Define  $x < y$  by  $f(x) <_{lex} f(y)$ . It is clear that  $<$  is computable in PF on  $L$ . Since lexicographic ordering is total on  $\Sigma^*$ ,  $<$  is total on  $L$ . Since  $f$  is p-invertible, it is honest, so we have  $|x| \leq \text{poly}(|f(x)|) \leq \text{poly}(|f(y)|) \leq$

$\text{poly}(|y|)$ , where the final inequality follows from the fact that  $f$  is computable in polynomial time.  $\square$

Next we note that  $p$ -cardinality preserves density up to polynomial transformations. Recall that the census function of a language  $L$  is  $c_L(n) = |L \cap \Sigma^{\leq n}|$ . We say that two functions  $f, g: \mathbb{N} \rightarrow \mathbb{N}$  are polynomially related if there are polynomials  $p, q$  such that  $f(n) \leq g(p(n))$  and  $g(n) \leq f(q(n))$  for all  $n$ .

**Lemma 4.23** *If  $\|A\|_p = \|B\|_p$ , then  $c_A(n)$  and  $c_B(n)$  are polynomially related.*

*Proof* Let  $f: A \rightarrow B$  be a  $p$ -equipollence witnessing  $\|A\|_p = \|B\|_p$ . Let  $p$  be a polynomial bounding the running time of  $f$  on inputs in  $A$ , and similarly let  $q$  be a polynomial bound on the running time of  $f^{-1}|_B$ . Then we have

$$f(A \cap \Sigma^{\leq n}) \subseteq B \cap \Sigma^{\leq p(n)} \quad \text{and} \quad f^{-1}(B \cap \Sigma^{\leq n}) \subseteq A \cap \Sigma^{\leq q(n)}.$$

The first implies  $c_A(n) \leq c_B(p(n))$  and the second implies  $c_B(n) \leq c_A(q(n))$ .  $\square$

**Corollary 4.24** *If  $L$  is  $p$ -countably infinite, then  $L$  is exponentially dense, that is,  $L$  has at least  $2^{\Omega(n^c)}$  strings of length  $\leq n$  for some  $c > 0$ .*

Since  $p$ -isomorphism preserves  $p$ -cardinality, but  $p$ -rankability is not  $p$ -isomorphism invariant, we recall Goldsmith & Homer's notion of scalability [26]. A language is  $p$ -scalable if it is  $p$ -isomorphic to a strongly  $p$ -rankable set. (We would like to introduce the obvious notion of weakly  $p$ -scalable and say that  $p$ -countability implies weak  $p$ -scalability, but we still do not know if the latter in fact holds!)

Combining the preceding results we get

**Theorem 4.25** *For any exponentially dense language  $L$ ,*

*strongly  $p$ -rankable  $\implies p$ -scalable  $\implies p$ -countable  $\implies$  weakly  $p$ -rankable w.r.t.  $<$ , for some length-related total ordering  $<$  on  $L$ , computable in PF.*

So exponential density is necessary to have the same  $p$ -cardinality as  $\Sigma^*$ , and among exponentially dense languages, having the  $p$ -cardinality of  $\Sigma^*$  sits in between strong and weak  $p$ -rankability (for some orderings). We believe the converses of these implications do not hold.

*Proof* We show the second arrow, the others having been shown above or previously. Goldberg & Sipser [25] noted that if a language is strongly  $p$ -rankable, then  $rk_L^{-1}: \Sigma^* \rightarrow L$  is computable by binary search in time polynomial in the length of its output. So it suffices to show that  $rk_L$  is honest. By assumption, we have  $rk_L(x) \geq 2^{|x|^c}$ . Taking bit-lengths of both sides we get  $|rk_L(x)| \geq |x|^c - 1$ , so  $|x| \leq |rk_L(x)|^{1/c}$ , and thus  $rk_L$  is honest.  $\square$

**Example 4.26** This allows us to give another interesting class of languages exhibiting a difference between  $p$ -isomorphism classes and  $p$ -cardinalities. Namely, any dense

proper subset of  $\Sigma^*$  that is p-scalable has the same p-cardinality as  $\Sigma^*$ , but is not isomorphic to  $\Sigma^*$ . The reason for the latter is that isomorphic sets have isomorphic complements, but the complement of  $\Sigma^*$  is empty, while that of the other language is non-empty by assumption.

#### 4.3.4 Finite Differences

We now show that finite differences preserve p-countability (for infinite sets, of course). The first part of this next result, on rankability, is surely known, but we could not find a reference, and its proof serves as a good warm-up for the second part; the second part we believe is new. We use the standard notation  $A =^* B$  to mean that  $A$  and  $B$  differ by at most a finite set.

**Theorem 4.27** *If  $A =^* B$ , then*

1.  *$A$  is strongly (resp., weakly) p-rankable iff  $B$  is;*
2.  *$A$  is p-countable iff  $B$  is.*

*Proof* Suppose  $A =^* B$ . If they are finite they are both strongly p-rankable and both p-countable, so we may now assume they are infinite.

(1) We will define a “shift function”  $\sigma(x)$  which tells us how much to shift as we move from  $A$  to  $B$ . We define  $\sigma$  inductively by

$$\sigma(x) = \sigma(x - 1) + B(x) - A(x)$$

where  $x - 1$  denotes the immediate predecessor of  $x$  in length-lexicographic order, and in case  $x = \epsilon$  (the empty string), we simply define by convention  $\sigma(\epsilon - 1) := 0$ .

Write  $B = (A \cup P) \setminus N$  where  $P$  is a finite set disjoint from  $A$ , and  $N$  is a finite subset of  $A$ . Let  $\{z_1, \dots, z_{s+t}\} = P \cup N$  with  $z_1 \leq_{lex} z_2 \leq_{lex} \dots \leq_{lex} z_{s+t}$ . First,  $\sigma$  is constant on each range  $[\epsilon, z_1), [z_1, z_2), \dots, [z_{s+t-1}, z_{s+t}), [z_{s+t}, \infty)$ . Next,  $\sigma$  can be computed in polynomial time: the  $z_i$  and the values of  $\sigma$  on each of the preceding ranges can be hardcoded, so that on input  $x$  an algorithm merely decides which of these  $O(1)$  ranges  $x$  is in, and then outputs the right (hard-coded) value. By construction, we have  $rk_B(x) = rk_A(x) + \sigma(x)$  for all strings  $x$ , giving the first statement.

(2) Suppose  $A$  is p-countably infinite (the finite case was handled in the first paragraph), and  $A =^* B$ . Let  $f: \mathbb{N} \rightarrow A$  be a p-equipollence. We will define a p-equipollence  $g: \mathbb{N} \rightarrow B$  by shifting  $f$ , similar to the above. Let  $B = (A \cup P) \setminus N$ , where  $P$  is a finite set disjoint from  $A$  and  $N$  is a finite subset of  $A$ .

First we show that  $B' = A \cup P$  is p-countable. The idea is to first enumerate  $P$ , then shift up the enumeration of  $A$  by  $|P|$ . More formally, we define  $g'(0), g'(1), \dots, g'(|P| - 1)$  to bijectively enumerate  $P$ , and then  $g'(i) = f(i - s)$  for all  $i \geq s$ . It is readily verified that  $g': \mathbb{N} \rightarrow B'$  is a p-equipollence (to see p-computability and p-invertibility, note that those first  $|P|$  values can be hard-coded).

Next, we show that  $B = B' \setminus N$  is p-countable. Let  $N = \{g'(j_1), \dots, g'(j_t)\}$  with  $j_1 < j_2 < \dots < j_t$ . The idea is to use the same p-equipollence  $g'$ , but to shift it at appropriate points to omit  $g'(j_1), \dots, g'(j_t)$ . Let  $h: \mathbb{N} \rightarrow \mathbb{N} \setminus \{j_1, \dots, j_t\}$  be the bijection such that  $h(i)$  is the  $i$ -th smallest element of  $\mathbb{N} \setminus \{j_1, \dots, j_t\}$ . It is readily verified that  $g = g' \circ h$  is an equipollence  $\mathbb{N} \rightarrow B$ . To see that it is p-computable and p-invertible, it suffices to show these for  $h$ , since  $g'$  is already a p-equipollence. For both of these, note that  $h$  is piecewise linear with only finitely many breakpoints. By hard-coding those breakpoints and their jump values (NB: it could jump more than one, if some of the  $j_k$  are consecutive integers),  $h$  is p-computable. Since  $h$  is also strictly increasing, computing its inverse amounts to figuring out which of the intervals of linearity the input is in, and then inverting the linear function on that interval. Again, as there are only finitely many such intervals, the necessary information can be hard-coded and the remaining computation is then easy.  $\square$

This raises the question of whether, for infinite sets, two sets that are finitely different must always be p-equipollent. We thank an anonymous reviewer for a suggested construction, a slight modification of which led us to the following resolution of this question in the negative:

**Theorem 4.28** *There is an infinite language  $A \in \mathcal{P}$  such that*

1.  *$A$  is not p-equipollent with any of its proper subsets (it is Dedekind p-finite, see Definition 6.2);*
2. *For any two languages  $B, B'$  that are both finitely different from  $A$ , we have  $\|B\|_p = \|B'\|_p$  if and only if  $|B \setminus A| - |A \setminus B| = |B' \setminus A| - |A \setminus B'|$ . In particular, not all sets finitely different from  $A$  are p-equipollent to  $A$ ; and*
3. *For any two languages  $B, B'$  that are both finitely different from  $A$ , we have  $\|B\|_p \leq \|B'\|_p$  iff  $|B \setminus A| - |A \setminus B| \leq |B' \setminus A| - |A \setminus B'|$ .*

In particular, for this  $A$ , the preorder  $\leq$  on the p-cardinals of sets finitely different from  $A$  is in fact a total order of the same order type as the integers. Although part (1) of the theorem already answers the question of whether finite differences preserve p-cardinality, the other two parts follow a similar proof, and will have an interesting consequence below (Corollary 4.48).

*Proof* Let  $A = \{1^{2^{2^k}} : k \in \mathbb{N}\}$ . Note that if  $\ell = 2^{2^k}$  is the length of a string in  $A$ , the next longest string has length exactly  $\ell^{\log_2 \ell}$  (this is why we used a tower of three exponentials; if we only use a tower of two exponentials, instead of  $\ell^{\log \ell}$ , we get  $\ell^2$  here).

(1) Suppose  $f: A \rightarrow A'$  is a p-equipollence to a proper subset  $A' \subsetneq A$ . Let  $n_0$  be such that for all  $n \geq n_0$ , the running time of  $f$  on strings of length  $\ell$  is strictly less than  $\ell^{\log \ell}$ . Then  $f$  cannot map any string in  $A$  of length at most  $n_0$  to a string of length  $> n_0$ , so we must have that  $f$  gives a bijection between  $A \cap \Sigma^{\leq n_0}$  and  $A' \cap \Sigma^{\leq n_0}$ , and since  $A' \subseteq A$ , in fact  $A'$  must agree with  $A$  on  $\Sigma^{\leq n_0}$ .

But on strings of length  $\geq n_0$ ,  $f$  must be length non-increasing, since for such a string,  $f(x)$  does not have time to write out any longer string in  $A$ . But since  $f$  was

already a bijection from the strings in  $A$  of length  $\leq n_0$  to themselves, any  $x \in A$  of length  $\geq n_0$  has nowhere to go but itself. Thus on strings of length  $\geq n_0$ ,  $f$  must be the identity. Therefore  $A' = A$ .

(2) Let  $B, B'$  be finitely different from  $A$ .

( $\Rightarrow$ ) Suppose  $\|B\|_p = \|B'\|_p$ ; let  $f : B \rightarrow B'$  be a  $p$ -equipollence. Let  $n_0$  denote one more than the maximum length of any string  $x$  such that  $B(x) \neq A(x)$  or  $B'(x) \neq A(x)$ . Note that, above length  $n_0$ ,  $B, B'$  agree with  $A$ , so they have the same  $\ell^{\log \ell}$  gap between their elements. Thus, as above, there exists some  $n_1 \geq n_0$  such that neither  $f$  nor  $f^{-1}$  can map any string of length  $\leq n_1$  to a string of length  $> n_1$ . But then  $f$  must map  $B \cap \Sigma^{\leq n_1}$  bijectively to  $B' \cap \Sigma^{\leq n_1}$  (with inverse  $f^{-1}$  restricted to  $B' \cap \Sigma^{\leq n_1}$ ).

Now, let  $P = B \setminus A, N = A \setminus B$ , and similarly define  $P', N'$  with  $B'$  in place of  $B$ . Then  $f|_{\Sigma^{\leq n_1}}$  gives a bijection from  $[(A \cup P) \setminus N] \cap \Sigma^{\leq n_1}$  to  $[(A \cup P') \setminus N'] \cap \Sigma^{\leq n_1}$ . As  $P, N, P', N' \subseteq \Sigma^{\leq n_1}$ —as part of the definition of  $n_1$ —and these are finite sets, and  $P, P'$  are disjoint from  $A$  and  $N, N'$  are subsets of  $A$ , we necessarily have  $|P| - |N| = |P'| - |N'|$ .

( $\Leftarrow$ ) Let  $n_0, P, P', N, N'$  be as above, and suppose  $|P| - |N| = |P'| - |N'|$ . The following map  $f$  is a  $p$ -equipollence  $B \rightarrow B'$ . On strings of length  $\leq n_0$ ,  $f$  implements an arbitrary, hard-coded bijection from  $[(A \cup P) \setminus N] \cap \Sigma^{\leq n_0}$  to  $[(A \cup P') \setminus N'] \cap \Sigma^{\leq n_0}$ , as these are both finite sets of the same cardinality. On strings of length  $> n_0$ ,  $f$  is the identity.

(3) Similar to (2). For the ( $\Rightarrow$ ) direction, let  $f : B \rightarrow B'' \subseteq B'$  be a  $p$ -equipollence. As above,  $f$  must be the identity map for all sufficiently long strings in  $B$ , say above some length  $n_1$ . But then  $f$  gives a bijection  $B \cap \Sigma^{\leq n_1} \rightarrow B'' \cap \Sigma^{\leq n_1} \subseteq B \cap \Sigma^{\leq n_1}$ . As before, counting these finite sets then tells us that  $|B \setminus A| - |A \setminus B| \leq |B' \setminus A| - |A \setminus B'|$ . For the ( $\Leftarrow$ ) direction, a construction similar to part (2) works, where the hard-coded part of  $f$  is now an injection rather than a bijection. □

**Open Question 4.29** *If  $A, B$  are infinite languages and  $A =^* B$  with at least  $n^c$  strings of length  $\leq n$  (for some  $c > 1$  and all sufficiently large  $n$ ), must  $\|A\|_p = \|B\|_p$ ? What if additionally  $A \subsetneq B$  with  $B \setminus A$  finite? What if, additionally,  $A \in \mathcal{P}$ ?*

Actually, the set in Theorem 4.28 does not even have logarithmic density!

### 4.4 Arithmetic of $p$ -cardinals

Since many versions of AC are about cardinal arithmetic, we develop arithmetic of  $p$ -cardinals.

**Proposition 4.30** ( $p$ -cardinal arithmetic) *If  $\|A\|_p = \|A'\|_p$  and  $\|B\|_p = \|B'\|_p$ , then  $\|A \oplus B\|_p = \|A' \oplus B'\|_p$  and  $\|A \times B\|_p = \|A' \times B'\|_p$ . If  $\|A\|_p \leq \|A'\|_p$  and  $\|B\|_p \leq \|B'\|_p$ , then  $\|A \oplus B\|_p \leq \|A' \oplus B'\|_p$  and  $\|A \times B\|_p \leq \|A' \times B'\|_p$ .*

Defining  $\|A\|_p + \|B\|_p = \|A \oplus B\|_p$  and  $\|A\|_p \times \|B\|_p := \|A \times B\|_p$ , the  $p$ -cardinals form a pre-ordered semi-ring with additive identity  $\|\emptyset\|_p$  and multiplicative identity  $\|\{\epsilon\}\|_p$ .

Furthermore:

1. For all  $A$ ,  $\|\emptyset\|_p \times \|A\|_p = \|\emptyset\|_p$ .
2. For all  $A$ ,  $\|\emptyset\|_p \leq \|A\|_p \leq \|\Sigma^*\|_p$ .
3. If  $A$  is infinite, then  $\|A\|_p > \|F\|_p$  for any finite set  $F$ .
4. For all  $A, B$ ,  $\|A\|_p \leq \|A\|_p + \|B\|_p$ .
5. For all  $A$  and for all nonempty  $B$ ,  $\|A\|_p \leq \|A\|_p \times \|B\|_p$ .
6. The  $p$ -cardinalities of finite sets form a totally ordered sub-semiring that is isomorphic as an ordered semiring to  $(\mathbb{N}, +, \times, \leq, 0, 1)$ .
7. For all finite non-empty languages  $n$  and all languages  $A$ , we have  $\|n \times A\|_p = \|A \oplus A \oplus \dots \oplus A\|_p$ , where the latter summation has  $|n|$  summands.

*Proof* Suppose  $f: A \rightarrow A''$  is a  $p$ -equipollence for some  $A'' \subseteq A'$  and  $g: B \rightarrow B''$  is a  $p$ -equipollence for some  $B'' \subseteq B'$ . Then  $f \oplus g$ , defined by

$$(f \oplus g)(bx) = \begin{cases} f(x) & \text{if } b = 0 \\ g(x) & \text{if } b = 1 \end{cases}$$

is a  $p$ -equipollence  $A \oplus B \rightarrow A'' \oplus B'' \subseteq A' \oplus B'$ , with inverse  $f^{-1} \oplus g^{-1}$ . Similarly,  $f \times g$ , defined by  $(f \times g)(x, y) = (f(x), g(y))$ , is a  $p$ -equipollence  $A \times B \rightarrow A'' \times B'' \subseteq A' \times B'$  with inverse  $f^{-1} \times g^{-1}$ . This gives us the desired statements for  $\leq$ ; since  $\leq$  is only a pre-order, the statements for  $=$  do not immediately follow. However, if in the proof we take  $A'' = A'$  and  $B'' = B'$  then we get the desired statements for  $=$ .

Finally, note that  $A \oplus \emptyset = \{0x : x \in A\}$ , which is easily seen to have the same  $p$ -cardinality as  $A$ . Similarly,  $A \times \{b\} = \{(a, b) : a \in A\}$  is again easily seen to have the same  $p$ -cardinality as  $A$ .

We leave the further properties as exercises for the reader.  $\square$

*Remark 4.31* Note that the arithmetic operations and comparisons are also well-defined on the  $\equiv$ -equivalence classes of  $p$ -cardinals (where  $\|A\|_p \equiv \|B\|_p$  if  $\|A\|_p \leq \|B\|_p$  and  $\|B\|_p \leq \|A\|_p$ ), in which case the result is a partially ordered semi-ring, not just pre-ordered.

**Observation 4.32**  $\|\Sigma^*\|_p = \|\Sigma^*\|_p + \|\Sigma^*\|_p$  and  $\|\Sigma^*\|_p = \|\Sigma^*\|_p \times \|\Sigma^*\|_p$ .

*Proof* Let  $n_x$  denote the natural number associated to  $x$  (see Preliminaries), and let  $\sigma_n$  denote the string associated to the natural number  $n$ . The bijection  $\Sigma^* \rightarrow \Sigma^* \oplus \Sigma^*$  is given by

$$f(x) = \begin{cases} 0\sigma_{n_x/2} & \text{if } n_x \text{ is even} \\ 1\sigma_{(n_x-1)/2} & \text{if } n_x \text{ is odd} \end{cases}$$

with inverse

$$f^{-1}(by) = \begin{cases} \sigma_{2n_y} & \text{if } b = 0 \\ \sigma_{2n_y+1} & \text{if } b = 1 \end{cases}$$



For the direct product, we use any of the standard bijections  $\Sigma^* \rightarrow \Sigma^* \times \Sigma^*$  (for example, using the bijections  $n, \sigma$  above to translate to  $\mathbb{N}$  and using the standard bijections  $\mathbb{N} \rightarrow \mathbb{N} \times \mathbb{N}$ ). □

**Corollary 4.33** *If  $A$  is  $p$ -countable, then*

$$A \cong^p \overline{A} \iff \overline{A} \text{ is } p\text{-countable,}$$

*and if these hold, then  $A \cong^p A \times \Sigma^*$  (i. e.,  $A$  is a  $p$ -cylinder).*

**Open Question 4.34** *What can be said if merely  $\|A\|_p = \|A \times \Sigma^*\|_p$  and  $\|\overline{A}\|_p = \|\overline{A} \times \Sigma^*\|_p$ ?*

*Proof* Suppose  $A$  is  $p$ -countable. The forward direction follows from the fact that  $p$ -isomorphism preserves  $p$ -countability. The backward direction follows from Corollary 4.19 using  $A = B$ .

For the final statement, suppose that both  $A, \overline{A}$  are  $p$ -countable. The idea of the proof is similar to that of Goldsmith, Kunen, and Hemaspaandra [27] and Corollary 4.19. Since  $\|A\|_p = \|\Sigma^*\|_p$  and  $\|\Sigma^*\|_p = \|\Sigma^*\|_p \times \|\Sigma^*\|_p$ , we may replace the LHS and one of the ones on the RHS with  $A$  to get  $\|A\|_p = \|A\|_p \times \|\Sigma^*\|_p = \|A \times \Sigma^*\|_p$ . Similarly for  $\overline{A}$ . Now let  $f: A \rightarrow A \times \Sigma^*$  and  $g: \overline{A} \rightarrow \overline{A} \times \Sigma^*$  be  $p$ -equipollences. Then the function

$$h(x) = \begin{cases} f(x) & x \in A \\ g(x) & x \notin A. \end{cases}$$

is a  $p$ -computable bijection  $\Sigma^* \rightarrow \Sigma^* \times \Sigma^*$ , since  $A \in P$ . To see that it is a bijection, note that  $\overline{A} \times \Sigma^* = \overline{A} \times \Sigma^*$ . The inverse of  $h$  is given similarly by the  $p$ -computable bijection

$$h^{-1}(y) = \begin{cases} f^{-1}(y) & y \in A \times \Sigma^* \\ g^{-1}(y) & y \notin A \times \Sigma^*. \end{cases}$$

Thus  $A \cong A \times \Sigma^*$ , so  $A$  is a cylinder. □

We note that the set of *all*  $p$ -cardinalities is uncountably infinite: there are uncountably many subsets of  $\Sigma^*$ , and each  $p$ -cardinality class has countably many representatives, since there are only countably many Turing machines (even ones that are only partial) to compute  $p$ -equipollences. However, even in interesting countable sets of  $p$ -cardinalities, we believe:

**Conjecture 4.35** For any standard complexity class  $\mathcal{C} \supseteq P$  (such as  $P, NP, PSPACE, EXP$ , etc.), the semiring of  $p$ -cardinalities of languages in  $\mathcal{C}$  is not finitely generated. The same for the semiring of equivalence class of  $p$ -cardinalities (see Remark 4.31).

For ordinary cardinals,  $\aleph_0$  is the smallest infinite cardinal, and every infinite cardinal  $\mathfrak{c}$  satisfies  $\mathfrak{c} + \aleph_0 = \mathfrak{c}$ . However, for  $p$ -cardinals this is no longer the case:

**Proposition 4.36** *If  $A$  is  $P$ -immune, then  $\|A\|_p \not\leq \|A\|_p + \|\Sigma^*\|_p$ .*

*Proof* Suppose  $\|A\|_p + \|\Sigma^*\|_p \leq \|A\|_p$ . Since the left-hand side is equal to  $\|A \oplus \Sigma^*\|_p$ , we have a  $p$ -equipollence  $f$  from  $A \oplus \Sigma^*$  to a subset  $A' \subseteq A$ . Then the restriction of  $f$  to  $1\Sigma^*$  gives a  $p$ -equipollence to a subset  $A'' \subseteq A' \subseteq A$ . Thus  $A''$  is  $p$ -countable, thus by Proposition 4.2 is in  $P$ . So  $A$  contains an infinite  $P$  subset, contradicting its  $P$ -immunity.  $\square$

**Proposition 4.37** (Non-cancellation of addition, Nerode & Rempel [50, Theorem 7]) *There exist languages  $A, B, C$  with  $\|A \oplus B\|_p = \|A \oplus C\|_p$  but  $\|B\|_p \neq \|C\|_p$ .*

Nerode & Rempel showed the result for unary languages, and using slightly different definitions, but their same construction works with our definitions.

Nerode & Rempel [50, Theorem 10] also showed that one does get cancellation of multiplication by finite sets of unary strings. Their argument used a back-and-forth construction that required them to compute their function on all predecessors of a string. Because they were using unary languages, there were only linearly many such predecessors and the entire argument could be carried out in polynomial time. When we attempt to do the same in our setting with, say, length-lexicographic ordering, a string has exponentially many predecessors (exponential in its length), so the same strategy doesn't work. It is possible that such a strategy could work with a  $p$ -well-founded ordering (see Definition 6.1) in which immediate predecessors were computable in FP, but we have not been able to get this to work, so we leave it as a question:

**Open Question 4.38** (*Division by finite  $p$ -cardinals*). *Is it the case that for any finite non-empty language  $n$  and for any languages  $A, B$ ,  $\|n \times A\|_p = \|n \times B\|_p$  implies  $\|A\|_p = \|B\|_p$ ?*

#### 4.5 Polynomial-Time Axioms of Choice based on $P$ -Cardinality

**Classical AC 4** (See [53, CN6 and CN7, pp. 52–53]) For any cardinals  $m < n$  and  $p < q$ ,  $m + p < n + q$  and  $mp < nq$ .

Compare to Proposition 4.30.

**Classical AC 5** (Tarski's Theorem [65], see [53, CN 3, p. 52]) For every infinite set  $A$ , there is a bijection between  $A$  and  $A \times A$ .

**Polynomial-Time AC (PAC) 4** For every infinite language  $L \in P$ ,  $\|L\|_p = \|L \times L\|_p$ .

**Classical AC 6** (Law of Trichotomy [53, T, p. 9]) For all sets  $A, B$ , either  $A$  is in bijection with a subset of  $B$  or  $B$  is in bijection with a subset of  $A$ .

A straightforward polynomial-time version of this is false by density considerations:

**Proposition 4.39** *There are two languages in  $\mathsf{P}$  whose  $p$ -cardinalities are not comparable.*

*Proof* Let  $\text{tow}(0) = 1$  and  $\text{tow}(n + 1) = 2^{\text{tow}(n)}$  be the tower function.  $A_0$  will consist of all strings  $x$  of length  $\text{tow}(2k) \leq |x| < \text{tow}(2k + 1)$  for all  $k \geq 0$ , and  $A_1$  will consist of all strings of length  $\text{tow}(2k + 1) \leq |x| < \text{tow}(2k + 2)$  for all  $k \geq 0$ . Clearly both  $A_0, A_1$  are in  $\mathsf{P}$ . The idea is that their densities fluctuate between  $\Theta(n)$  and  $2^{\Theta(n)}$  infinitely often, but at opposite times. Let us verify this.

For lengths in the range  $[\frac{1}{2}\text{tow}(2k + 2), \text{tow}(2k + 2))$ , all the strings in  $A_0$  have length  $< \text{tow}(2k + 1)$ , so there are at most  $2^{\text{tow}(2k+1)+1} - 1 \sim \text{tow}(2k + 2)$  of them. So in these ranges we have  $c_{A_0}(n) \in [n/2, n]$ . In this same range, all strings are present in  $A_1$ , so we have  $c_{A_1}(n) \geq 2^n$ , which is nearly maximal. For lengths in the range  $[\frac{1}{2}\text{tow}(2k + 1), \text{tow}(2k + 1))$  the situation is reversed.

For the sake of contradiction, suppose  $B \subseteq A_0$  has  $\|B\|_p = \|A_1\|_p$ . Since  $A_0$  is empty at lengths in the ranges  $[\text{tow}(2k + 1), \text{tow}(2k + 2))$ , so is  $B$ , and thus  $B$  has density at most  $O(n)$  for  $n$  in the range  $[\frac{1}{2}\text{tow}(2k + 2), \text{tow}(2k + 2)]$ . But  $A_1$  has density at least  $2^n$  in this range, so the two are not polynomially related for arbitrarily large stretches of lengths, a contradiction. The analogous argument swapping the roles of  $A_0$  and  $A_1$  and using the ranges  $[\text{tow}(2k), \text{tow}(2k + 1))$  instead rules out a subset of  $A_1$  having the same  $p$ -cardinality as  $A_0$ . Thus  $\|A_0\|_p$  and  $\|A_1\|_p$  are incomparable.  $\square$

We thus propose a polynomial-time form of AC with density in its assumption.

**Polynomial-Time AC (PAC) 5** For all sets  $A, B \in \mathsf{P}$  of polynomially related densities, either  $\|A\|_p \leq \|B\|_p$  or  $\|B\|_p \leq \|A\|_p$ .

If we remove the restriction that both languages are in  $\mathsf{P}$ , we believe the statement becomes false.

**Classical AC 7** (Law of Trichotomy [53, T', p. 10]) For every two non-empty sets, there is a surjective mapping of one onto the other.

**Polynomial-Time AC (PAC) 6** For every two non-empty languages  $A, B$  with  $c_A(n) \leq c_B(\text{poly}(n))$ , there is an honest surjective polynomial-time function  $A \rightarrow B$ .

Classically, T implies T'. For the  $p$ -analogues, such an implication seems tantamount to Hypothesis Q, though formalizing this has proved tricky. We do have, however:

**Classical AC 8** Every surjective function has a right inverse.

**Polynomial-Time AC (PAC) 7** Every polynomial-time computable, honest, surjective function has a polynomial-time inverse. This is an exact restatement of Hypothesis Q.

**Classical AC 9** (See [53, Section 6, pp. 52–53]) Cardinal forms of the axiom of choice. Throughout,  $m, n, p, q$  denote infinite cardinals. When not quantified, uni-

versal quantification is assumed, e. g., the first condition is more precisely “For all infinite cardinals  $m, n, m \cdot n = m + n$ .”

1.  $m \cdot n = m + n$ .
2. There is a cardinal  $n$  such that  $m = n^2$ .
3. If  $m^2 = n^2$  then  $m = n$ .
4. If  $m + p < n + p$  then  $m < n$ .
5. If  $mp < np$  then  $m < n$ .
6. Every cardinal has an immediate successor ( $m < n$  and if  $m < p$  then  $n \leq p$ ).
7. If  $n < p$  and there is no cardinal between  $n$  and  $p$  ( $p$  “covers”  $n$ ) then either  $mn = mp$  or  $mp$  covers  $mn$ . (If  $p$  is an immediate successor of  $n$  then  $p$  covers  $n$ . The converse is equivalent to AC.)
8. If  $m < n$  then there is a  $p$  such that  $n = mp$ .
9. If  $m < n$  then  $n/m$  exists (as in the previous point) and is unique.
10. If  $m < n$  then  $n/m = n$ .
11. If  $m + p = m + q$  then either  $p = q$ , or  $p \leq m$  and  $q \leq m$ .
12. If  $m + m < m + n$  then  $m < n$ . (The converse is independent of AC.)
13. If  $m < n$  then  $n - m$  exists, that is, there exist one and only one  $p$  such that  $n = m + p$ . (Existence alone is independent of AC.)
14. If  $m < n$  then  $n - m = n$ .
15. If  $p < n, q < n$  then  $p + q \neq n$ .
16. If  $p < n, q < n$  then  $pq \neq n$ .
17. Either  $mn = m$  or  $mn = n$ .

Many of the arguments that these are equivalent to one another and to the standard AC rely on the notion of the immediate successor of a cardinal (e. g., [53]). We show that such a construction is unlikely to exist for (infinite)  $p$ -cardinalities, by a Ladner-type diagonalization result.

Before coming to the result, our proof currently uses one additional assumption, essentially because of the issue raised by Theorem 4.28.

**Definition 4.40** (Much smaller  $p$ -cardinality) We say that  $\|A\|_p$  is *much smaller* than  $\|B\|_p$ , denoted  $\|A\|_p \ll \|B\|_p$ , if  $\|A\|_p < \|B\|_p$  and for any subset  $B' \subseteq B$  such that  $\|A\|_p = \|B'\|_p$ , the difference  $B \setminus B'$  is infinite.

The next proposition shows that  $\ll$  is in fact a well-defined relationship on  $p$ -cardinality classes.

**Proposition 4.41** *If  $\|A\|_p = \|\hat{A}\|_p$  and  $\|B\|_p = \|\hat{B}\|_p$ , then  $\|A\|_p \ll \|B\|_p$  iff  $\|\hat{A}\|_p \ll \|\hat{B}\|_p$ .*

*Proof* Suppose  $\|A\|_p \ll \|B\|_p$ , we show that  $\|\hat{A}\|_p \ll \|\hat{B}\|_p$ , with the reverse implication following by symmetry. If  $\|A\|_p = \|\hat{A}\|_p$ , then for any subset  $B' \subseteq B$ , we have  $\|A\|_p = \|B'\|_p$  iff  $\|\hat{A}\|_p = \|B'\|_p$ , so we have  $\|\hat{A}\|_p \ll \|B\|_p$ . Now suppose  $\|B\|_p = \|\hat{B}\|_p$ , and let  $\hat{B}' \subseteq \hat{B}$  have the same  $p$ -cardinality as  $\hat{A}$ . Let  $f: \hat{B} \rightarrow B$  be a  $p$ -equipollence, and define  $B' = f(\hat{B}')$ . Then we have  $\|A\|_p = \|B'\|_p$ , and by assumption  $B \setminus B'$  is infinite. But then  $f^{-1}(B \setminus B') = \hat{B} \setminus \hat{B}'$  is infinite as well, since  $f^{-1}$  is injective on  $B$ .  $\square$

Obviously finite sets are much smaller than  $\|A\|_p$  for any infinite  $A$ , but we also have less trivial examples.

**Proposition 4.42** 1. If  $A \subset B$  and  $A \not\equiv_m^p B$ , then  $\|A\|_p \ll \|B\|_p$ .

2. If  $A \not\equiv_m^p B \neq \Sigma^*$ , then  $\|A\|_p \ll \|\Sigma^* \oplus B\|_p$ .

*Proof* 1. If  $A \subset B$  then  $\|A\|_p \leq \|B\|_p$ . If  $A \not\equiv_m^p B$ , suppose  $B' \subset B$  has the same  $p$ -cardinality as  $A$ . Since  $A, B'$  are both proper subsets of  $B$ , neither can be  $\Sigma^*$ , so Proposition 4.2 implies  $A \equiv_m^p B'$ . But since  $B \not\equiv_m^p A$ , we must have  $B \not\equiv_m^p B'$ , and thus  $B \setminus B'$  must be infinite.

2. Define  $\hat{B} = \Sigma^* \oplus B$  and  $\hat{A} = 0A$ . As  $\hat{B} \equiv_m^p B$  and  $\hat{A} \equiv_m^p A$ , we still have  $\hat{A} \not\equiv_m^p \hat{B}$ , and now we have  $\hat{A} \subseteq \hat{B}$ . Part 1 implies  $\|\hat{A}\|_p \ll \|\hat{B}\|_p$ , and Proposition 4.41 then gives  $\|A\|_p \ll \|\hat{B}\|_p$  as well. □

**Warning!** “Much smaller” is, unfortunately, not transitive. Suppose  $A \not\equiv_m^p B \neq \Sigma^*$ . Applying part 2 of the preceding proposition a few times, we get  $\|A\|_p \ll \|\Sigma^* \oplus B\|_p \ll \|\Sigma^* \oplus A\|_p \ll \|\Sigma^* \oplus \Sigma^* \oplus B\|_p = \|\Sigma^* \oplus B\|_p$ . But obviously  $\|\Sigma^* \oplus B\|_p$  is equal to itself, so it cannot be much less than itself. At first sight this seems to point to some error, but it is actually a natural consequence of the fact that  $p$ -cardinality mixes subsets and many-one degrees, while the poset of subset inclusion is not compatible with the poset of many-one reductions. Viz., there are sequences of languages  $L_1 \subset L_2 \subset L_3$  with  $L_2$  NP-complete but  $L_1, L_3 \in P$ , e. g.,  $0L_1 \subseteq \Sigma^* \oplus SAT \subseteq \Sigma^*$  for any  $L_1 \in P$ .

In contrast, the relation “ $\leq$  but not  $\ll$ ” (“smaller, but not much smaller”) is transitive. For if  $A$  is  $p$ -equipollent to a cofinite subset of  $B$ , and  $B$  is  $p$ -equipollent to a cofinite subset of  $C$ , then composing the two gives a  $p$ -equipollence between  $A$  and a cofinite subset of  $C$ . We will use this fact (in contrapositive form) several times in the following proof.

**Theorem 4.43** Let  $A, B$  be languages with  $\|A\|_p \ll \|B\|_p$ . Then there exists an infinite language  $C$  with  $\|A\|_p < \|C\|_p < \|B\|_p$ .

Because of Theorem 4.28, the assumption here is not equivalent to  $\|A\|_p < \|B\|_p$ ; we will see in Corollary 4.48 below that the same set constructed in Theorem 4.28 can be used to give an example of sets  $A, B$  with  $\|A\|_p < \|B\|_p$  but with no  $p$ -cardinal strictly in between the two. In the vein of Question 4.29, it is interesting to ask for conditions under which Theorem 4.43 would hold with the weaker assumption  $\|A\|_p < \|B\|_p$ , and/or with the stronger conclusion  $\|A\|_p \ll \|C\|_p \ll \|B\|_p$ . As is, we necessarily have that at least one of  $\|A\|_p < \|C\|_p$  and  $\|C\|_p < \|B\|_p$  can be turned into  $\ll$ , for otherwise we could not have  $\|A\|_p \ll \|B\|_p$ , but our construction does not control which one.

*Proof* By assumption, there is a  $p$ -computable,  $p$ -invertible bijection from  $A$  to a cofinite subset of  $B$ , but not vice versa. (Note that  $B$  cannot be finite.) To simplify notation in what follows, let us identify  $A$  with its image under this bijection, so that, without loss of generality, we may assume that in fact  $A \subsetneq B$  and  $B \setminus A$  is infinite. We will build  $C$  such that  $A \subsetneq C \subsetneq B$  and  $\|A\|_p < \|C\|_p < \|B\|_p$ . Since  $A \subseteq C \subseteq B$ , we have  $\|A\|_p \leq \|C\|_p \leq \|B\|_p$ . So it suffices to build  $C$  in between  $A$  and  $B$  avoiding  $p$ -computable,  $p$ -invertible bijections  $C \rightarrow A$  and  $B \rightarrow C$ .

Let  $\hat{M}_1, \hat{M}_2, \hat{M}_3, \dots$  be an enumeration of Turing machine transducers, and for all  $c$  let  $M_c$  be the machine gotten from  $\hat{M}_c$  by restricting  $\hat{M}_c$  to run on each input of length  $n$  for no more than  $cn^c + c$  steps. If  $M_c(x)$  has not made an output after  $c|x|^c + c$  steps, then  $M_c(x)$  rejects  $x$  (i. e., it makes no output). It is clear that  $M_1, M_2, \dots$  is thus an enumeration of all partial polynomial-time Turing machine transducers.

**Construction.** We build  $C$  in stages  $s = 0, 1, 2, \dots$ . We also build a set  $E$  of strings to be excluded from  $C$ . For each  $s \in \mathbb{N}$ ,  $C_s$  and  $E_s$  will be the parts of  $C$  and  $E$ , respectively, enumerated at the end of stage  $s$ . The construction will guarantee that, for each  $s$ ,  $C_s \cap E_s = \emptyset$ ,  $A \subseteq C_s \subseteq C_{s+1} \subseteq B$ ,  $C_s$  will be finitely different from  $A$ ,  $E_s \subseteq E_{s+1} \subseteq B$ , and  $E_s$  will be finite.

We will have two sets of requirements to satisfy below; when  $s \equiv 0 \pmod{3}$ , we will add to  $C_s$  to ensure that  $C$  is infinite. The stages  $s + 1$  where  $s \equiv 1 \pmod{3}$  (resp.  $2 \pmod{3}$ ) will ensure the first (resp., second) set of requirements are satisfied at this stage. We break the construction into numbered cases for reference in the verification below.

*Stage  $s = 0$ .* Set  $C_0 = A$  and  $E_0 = \emptyset$ .

*Stage  $s + 1$ , where  $s \equiv 0 \pmod{3}$ .* Let  $y$  be the least element of  $B \setminus (C_s \cup E_s)$  (any element will do), and set  $C_{s+1} := C_s \cup \{y\}$  and  $E_{s+1} := E_s$ . (Such a  $y$  must exist since  $B \setminus A$  is infinite,  $C_s$  is finitely different from  $A$ , and  $E_s$  is finite.)

*Stage  $s + 1$ , where  $s = 3\langle\alpha, \beta\rangle + 1$ .* (Here we use  $\langle\bullet, \bullet\rangle$  to denote a bijection  $\mathbb{N} \times \mathbb{N} \rightarrow \mathbb{N}$ .)

Case 1:  $M_\alpha|_A: A \rightarrow M_\alpha(A)$  is not a p-equipollence, or  $M_\alpha(A) \not\subseteq B$ , or  $M_\beta$  is not its inverse p-equipollence  $M_\alpha(A) \rightarrow A$ . In this case, set  $C_{s+1} := C_s$  and  $E_{s+1} := E_s$ , and continue to the next stage ( $s + 2$ ).

Case 2: Otherwise. Note that in this case  $B \setminus M_\alpha(A)$  must be infinite, since  $\|A\|_p = \|M_\alpha(A)\|_p$ , but  $\|A\|_p \ll \|B\|_p$ . Let  $Y = B \setminus E_s$ ; since  $E_s$  is finite,  $Y$  is cofinite in  $B$ . Since  $\|A\|_p \ll \|B\|_p$  by assumption, but  $Y$  is cofinite in  $B$ ,  $\|A\|_p$  cannot be equal to  $\|Y\|_p$ .

Subcase 2.1:  $Y \not\subseteq \text{dom}(M_\beta)$ . In this subcase, let  $y$  be the least element of  $Y \setminus \text{dom}(M_\beta)$  (any element will do), set  $C_{s+1} := C_s \cup \{y\}$ ,  $E_{s+1} := E_s$ , and go to the next stage.

Subcase 2.2: Not in the previous subcase, and  $M_\beta$  is not injective on  $Y$ . In this subcase, let  $y_1, y_2 \in Y$  be the least two distinct elements that  $M_\beta$  maps to the same place (any two will do). Set  $C_{s+1} := C_s \cup \{y_1, y_2\}$  and  $E_{s+1} := E_s$  and continue to the next stage.

Subcase 2.3: Not in the previous subcases, and  $M_\beta(Y) \not\subseteq A$ . In this subcase, let  $y$  be the least element of  $Y$  such that  $M_\beta(y) \notin A$  (any will do), set  $C_{s+1} := C_s \cup \{y\}$ ,  $E_{s+1} := E_s$ , and go to the next stage.

Subcase 2.4: Not in the previous subcases. In this subcase, there must exist  $y \in Y$  such that  $M_\alpha(M_\beta(y)) \neq y$ ; for at this point we have that  $M_\beta|_Y$  is an injective function from  $Y$  into  $A$ , so if  $M_\alpha$  were its inverse we would have  $\|A\|_p = \|Y\|_p$  (for we already know that  $(M_\beta \circ M_\alpha)|_A = id_A$ ), which we showed above was impossible. Let  $y$  be the least such (any such

will do), set  $C_{s+1} := C_s \cup \{y\}$ ,  $E_{s+1} := E_s$ , and go to the next stage.

Stage  $s + 1$  where  $s = 3(\alpha, \beta) + 2$ .

Case 1: In any of the following subcases, set  $C_{s+1} := C_s$ ,  $E_{s+1} := E_s$ , and go to the next stage.

Subcase 1.1:  $M_\alpha(C_s) \not\subseteq B$ .

Subcase 1.2: It is not the case that  $M_\alpha|_{C_s} : C_s \rightarrow M_\alpha(C_s)$  is a p-equipollence with inverse  $M_\beta$ .

Subcase 1.3:  $X := B \setminus (M_\alpha(C_s) \cup E_s)$  is not contained in  $\text{dom}(M_\beta)$ .

Subcase 1.4: Not in the above subcases, and  $M_\beta(X) \subseteq C_s$ .

Case 2: Not in any of the above subcases. In this case, let  $x$  be the least element of  $X$  such that  $M_\beta(x) \notin C_s$  (any such will do), set  $E_{s+1} := E_s \cup \{x\}$ ,  $C_{s+1} := C_s$ , and go to the next stage. (Note that such an  $x$  must exist, for  $E_s$  is finite, and  $M_\alpha$  is a p-equipollence from a set that is finitely different from  $A$  to a subset of  $B$ , but  $\|A\|_p \ll \|B\|_p$ .)

This completes the description of stage  $s + 1$ , and thus of the construction.

**Requirements.** Below, we will show that, in addition to ensuring  $C$  is infinite, the construction satisfies two sets of requirements, and that these requirements imply the desired property of  $C$ . For the purposes of stating these requirements, define  $E = \bigcup_s E_s = \lim_{s \rightarrow \infty} E_s$ . The requirements for  $(\alpha, \beta)$  are as follows:

$R1_{\alpha,\beta}$ : At least one of the following holds

- (a)  $A \not\subseteq \text{dom}(M_\alpha)$  or  $M_\alpha$  is not injective on  $A$ , or
- (b)  $C \not\subseteq \text{dom}(M_\beta)$  or  $M_\beta$  is not injective on  $C$ , or
- (c)  $M_\alpha(A)$  is not contained in  $B$  (sic!), or
- (d)  $M_\beta(C)$  is not contained in  $A$ , or
- (e) there is  $a \in A$  with  $M_\beta(M_\alpha(a)) \neq a$  or there is  $c \in C$  with  $M_\alpha(M_\beta(c)) \neq c$ .

$R2_{\alpha,\beta}$ : At least one of the following holds

- (a)  $C \not\subseteq \text{dom}(M_\alpha)$  or  $M_\alpha$  is not injective on  $C$ , or
- (b)  $B \not\subseteq \text{dom}(M_\beta)$  or  $M_\beta$  is not injective on  $B$ , or
- (c)  $M_\alpha(C)$  is not contained in  $B$ , or
- (d)  $M_\beta(B) \cap E$  is nonempty, or
- (e) there is  $c \in C$  with  $M_\beta(M_\alpha(c)) \neq c$  or there is  $b \in B$  with  $M_\alpha(M_\beta(b)) \neq b$ .

**The requirements suffice.** First let us see that the requirements, plus  $C$  being infinite, suffice for the theorem. The requirement  $R1_{\alpha,\beta}$  implies that  $M_\alpha$  is not a p-equipollence  $A \rightarrow C$  with inverse  $M_\beta$ , and  $R2_{\alpha,\beta}$  implies that  $M_\alpha$  is not a p-equipollence  $C \rightarrow B$  with inverse  $M_\beta$  (note that  $R2_{\alpha,\beta}$ (d) ensures  $M_\beta(B) \not\subseteq C$ , since  $E$  is disjoint from  $C$ ). Since  $(\alpha, \beta)$  run over a complete list of polynomial-time Turing machines, these requirements will establish that  $\|A\|_p < \|C\|_p < \|B\|_p$ .

The stages with  $s + 1 \equiv 0 \pmod{3}$  add elements to  $C$  infinitely often, ensuring that  $C$  is infinite even if  $A$  was finite.

**Verification of the requirements.** We begin by noting that this is an “injury-free” argument, in the sense that once a requirement is satisfied, it never becomes unsatisfied (“is never injured”). This is because each requirement is a  $(\Sigma_1^0)^{A \oplus B}$

statement, that is, it is of the form  $(\exists \vec{x})[P(\vec{x})]$  where  $P$  is a predicate that is computable with an oracle for  $A$  and  $B$ . Thus, it suffices to show that at the end of stage  $s + 1 = 3\langle \alpha, \beta \rangle + b$ , if  $C', E'$  are any two sets satisfying  $C_s \subseteq C' \subseteq B$ ,  $E_s \subseteq E' \subseteq B \setminus C'$ , then  $Rb_{\alpha, \beta}$  is satisfied with  $C'$  (resp.,  $E'$ ) in place of  $C$  (resp.,  $E$ ).

*Remark 4.44* To ensure this  $(\Sigma_1^0)^{A \oplus B}$  property, we note that  $R1_{\alpha, \beta}$  is slightly stronger than is needed to ensure  $M_\alpha$  is not a p-equipollence  $A \rightarrow C$  with inverse  $M_\beta$ , and similarly  $R2_{\alpha, \beta}$  is slightly stronger than is needed to ensure  $M_\alpha$  is not a p-equipollence  $C \rightarrow B$  with inverse  $M_\beta$ . In particular, if  $R1_{\alpha, \beta}(c)$  were merely “ $M_\alpha(A) \neq C$ ” and  $R1_{\alpha, \beta}(d)$  were “ $M_\beta(C) \neq A$ ,” it would suffice to avoid p-equipollences, but then these statements would not be  $\Sigma_1^0$  in  $A \oplus B$ , for it would then be possible that later additions to  $C$  would violate such a condition. Similarly, if  $R2_{\alpha, \beta}(c)$  were merely “ $M_\alpha(C) \neq B$ ” and  $R2_{\alpha, \beta}(d)$  were merely “ $M_\beta(B) \neq C$ ,” it would suffice to avoid p-equipollences, but again, would not so clearly avoid injuries from future additions to  $C$ .

*Stage  $s + 1$  with  $s = 3\langle \alpha, \beta \rangle + 1$  ensures  $R1_{\alpha, \beta}$  is satisfied.* We break the verification into cases according to the cases in the construction.

Case 1. If  $M_\alpha|_A : A \rightarrow M_\alpha(A)$  is not a p-equipollence to a subset  $M_\alpha(A) \subseteq B$  with inverse  $M_\beta$ , then Case 1 of the construction tells us to move onto the next stage. We must thus show that in this case, without any further changes to  $C_s, E_s$ , it is already the case that R1 is satisfied.

In this case, one of the following must hold:

- (A)  $A \not\subseteq \text{dom}(M_\alpha)$  or  $M_\alpha$  is not injective on  $A$ .
- (B)  $M_\alpha(A) \not\subseteq B$ .
- (C)  $M_\alpha(A) \not\subseteq \text{dom}(M_\beta)$
- (D)  $M_\beta$  is not injective on  $M_\alpha(A)$
- (E)  $M_\beta(M_\alpha(A)) \not\subseteq B$
- (F)  $M_\alpha$  and  $M_\beta$  are not inverses on  $A, M_\alpha(A)$ , respectively. That is, there is some  $a \in A$  such that  $M_\beta(M_\alpha(a)) \neq a$  or some  $b \in M_\alpha(A)$  such that  $M_\alpha(M_\beta(b)) \neq b$ . The latter is equivalent to the existence of an  $a \in A$  such that  $M_\alpha(M_\beta(M_\alpha(a))) \neq M_\alpha(a)$ .

Suppose (A) holds. (A) is the same as  $R1_{\alpha, \beta}(a)$ , so  $R1_{\alpha, \beta}$  is satisfied.

Suppose (B) holds. (B) is the same as  $R1_{\alpha, \beta}(c)$ .

Suppose (C) holds. Then there is an  $a \in A$  such that  $M_\alpha(a) \notin \text{dom}(M_\beta)$  and thus  $M_\beta(M_\alpha(a)) \neq a$  since the LHS is undefined, so  $R1_{\alpha, \beta}(e)$  holds.

Suppose (D) holds. Then there are distinct  $a_1, a_2 \in A$  such that  $M_\beta(M_\alpha(a_1)) = M_\beta(M_\alpha(a_2))$ . But then at least one of  $M_\beta(M_\alpha(a_i)) = a_i$  for  $i = 1, 2$  must fail, so  $R1_{\alpha, \beta}(e)$  holds.

Suppose (E) holds. Since  $M_\beta(M_\alpha(A)) \not\subseteq B$ , there is some  $a \in A$  such that  $M_\beta(M_\alpha(a)) \notin B \supseteq A$ , so we must have  $M_\beta(M_\alpha(a)) \neq a$ , so  $R1_{\alpha, \beta}(e)$  holds.

Suppose (F) holds. Then it must be the case that there is an  $a \in A$  such that  $M_\beta(M_\alpha(a)) \neq a$ . For if not, then by the second part of (F) we get that



there is an  $a$  such that

$$M_\alpha(M_\beta(M_\alpha(a))) \neq M_\alpha(a).$$

But if  $M_\beta(M_\alpha(a)) = a$  for all  $a \in A$ , then the displayed equation simplifies to  $M_\alpha(a) \neq M_\alpha(a)$ , which is absurd. Thus  $R1_{\alpha,\beta}(e)$  is satisfied.

Thus, case 1 ensures  $R1_{\alpha,\beta}$ , as claimed.

Case 2. Suppose instead we are in case 2, in which  $M_\alpha|_A : A \rightarrow M_\alpha(A)$  is a p-equipollence to a subset of  $B$  with inverse  $M_\beta$ . Following the notation in the construction, let  $Y = B \setminus E_s$ .

Subcase 2.1. If the condition of subcase 2.1 is satisfied ( $Y \not\subseteq \text{dom}(M_\beta)$ ), then we add a  $y \in Y \setminus \text{dom}(M_\beta)$  to  $C$ , thus satisfying  $R1_{\alpha,\beta}(b)$ .

Subcase 2.2. In subcase 2.2, if the relevant condition is satisfied, we add to  $C$  distinct  $y_1, y_2 \in Y$  such that  $M_\beta(y_1) = M_\beta(y_2)$ . This satisfies the second part of  $R1_{\alpha,\beta}(b)$ .

Subcase 2.3. In subcase 2.3, if the relevant condition is satisfied, we add to  $C$  a  $y \in Y$  such that  $M_\beta(y) \notin A$ , thus satisfying  $R1_{\alpha,\beta}(d)$ .

Subcase 2.4. In subcase 2.4, if the relevant condition is satisfied, we add to  $C$  a  $y \in Y$  such that  $M_\alpha(M_\beta(y)) \neq y$ , thus satisfying the second part of  $R1_{\alpha,\beta}(e)$ .

Since the cases and subcases exhaust all possibilities (case 2 is the “otherwise” of case 1, and subcase 2.4 captures all cases not in the previous cases by construction), this stage ensures that  $R1_{\alpha,\beta}$  is satisfied.

Stage  $s+1 = 3(\alpha, \beta)+2$  ensures  $R2_{\alpha,\beta}$ . Following the notation in the construction, let  $X = B \setminus (M_\alpha(C_s) \cup E_s)$ .

Case 1. We must show that in this case, without any further changes to  $C_s, E_s$ , it is already the case that  $R2$  is satisfied.

Subcase 1.1:  $M_\alpha(C_s) \not\subseteq B$ . Thus we have that  $M_\alpha(C) \not\subseteq B$ , which is precisely  $R2_{\alpha,\beta}(c)$ .

Subcase 1.2:  $M_\alpha|_{C_s} : C_s \rightarrow M_\alpha(C_s)$  is not a p-equipollence with inverse  $M_\beta$ . In this case, one of  $R2_{\alpha,\beta}(a), (b),$  or  $(e)$  must be satisfied (since together they are the definition of a p-equipollence), and thus  $R2_{\alpha,\beta}$  is satisfied already.

Subcase 1.3:  $X \not\subseteq \text{dom}(M_\beta)$ . Since  $X \subseteq B$  by definition,  $R2_{\alpha,\beta}(b)$  is satisfied.

Subcase 1.4: Not in the above subcases, and  $M_\beta(X) \subseteq C_s$ . In this case, we claim that  $M_\beta$  cannot be injective on  $B$ . To see that  $M_\beta$  is not injective on  $B$ , first note that since we are not in subcase 1.2, we have that  $M_\beta$  is the p-equipollence  $M_\alpha(C_s) \rightarrow C_s$  that is inverse to  $M_\alpha|_{C_s}$ . Thus  $M_\beta$  maps  $M_\alpha(C_s)$  bijectively onto  $C_s$ . Now let  $x \in X$ , and let  $b = M_\alpha(M_\beta(x))$ . Since  $M_\beta(X) \subseteq C_s$  by assumption, we have  $b \in M_\alpha(C_s)$ . But  $X$  is disjoint from  $M_\alpha(C_s)$  by definition, so  $b \neq x$ .

But since  $M_\beta, M_\alpha$  are inverses on  $M_\alpha(C_s), C_s$  respectively, and  $M_\beta(X) \subseteq C_s$  by assumption, we have  $M_\beta(b) = M_\beta(M_\alpha(M_\beta(x))) = (M_\beta \circ M_\alpha)(M_\beta(x)) = (id)(M_\beta(x)) = M_\beta(x)$ . Thus  $M_\beta$  is not injective on  $B$ , satisfying the second part of  $R2_{\alpha,\beta}(b)$ .

Case 2. In this case,  $E_s$  gets extended to  $E_{s+1}$  by adding some  $x \in X$  such that  $M_\beta(x) \notin C_s$ . Since  $X \subseteq B$ , this ensures that  $M_\beta(B) \cap E$  is not empty, satisfying  $R2_{\alpha,\beta}(d)$ .

Since the cases and subcases exhaust all possibilities, at the end of this stage the requirements  $R2_{\alpha,\beta}$  are satisfied.

Putting these all together, the construction indeed produces an infinite set  $C$  such that  $\|A\|_p < \|C\|_p < \|B\|_p$ , completing the proof of the theorem.  $\square$

*Remark 4.45* In addition to R1 and R2 being stronger than needed (see Remark 4.44), the reader may also wonder why R1 and R2 are not perfect analogues of one another. The answer is that we actually first wrote a proof with them being more symmetric, then pared them down to the only parts of the requirements we actually ended up needing.

Finally, we note that the argument is *almost* monotone in  $C$ . The only requirement that is not monotone in  $C$  is  $R2_{\alpha,\beta}(d)$ , and it was this requirement that forced us to keep track of the set  $E$  of excluded strings ( $R2_{\alpha,\beta}(d)$  is monotone in  $E$ ).

Given that  $\aleph_0$  is the unique minimum infinite cardinal, it is natural to wonder whether minimal infinite  $p$ -cardinals exist. The following corollary answers this in the negative:

**Corollary 4.46** (No minimal infinite  $p$ -cardinals) *For every infinite  $B \subseteq \Sigma^*$ , there is an infinite set  $C$  such that  $\|C\|_p < \|B\|_p$ .*

*Proof* Apply Theorem 4.43 with  $A = \emptyset$ .  $\square$

We suspect a modification of the above proof could be used to answer the following stronger question in the negative:

**Open Question 4.47.** Do there exist  $p$ -cardinals that are minimal(ly infinite) under  $\equiv$ ? That is, an infinite language  $B \subseteq \Sigma^*$  such that if  $\|C\|_p < \|B\|_p$ , then either  $C$  is finite or  $\|B\|_p \leq \|C\|_p$ ?

We conclude this section by showing that the assumption  $\|A\|_p \ll \|B\|_p$  in Theorem 4.43 cannot, in general, be replaced by  $\|A\|_p < \|B\|_p$ , even for infinite languages (for finite languages this is clear, whenever  $|B| = |A| + 1$ ), by the following corollary to Theorem 4.28:

**Corollary 4.47** (Infinite  $p$ -cardinals can have immediate predecessors) *There exist infinite languages  $A, B$  with  $\|A\|_p < \|B\|_p$  such that any language  $C$  with  $\|A\|_p \leq \|C\|_p \leq \|B\|_p$  is  $p$ -equipollent to either  $A$  or  $B$ .*

By Theorem 4.43, necessarily any such pair cannot have  $\|A\|_p \ll \|B\|_p$ .

*Proof* Let  $A$  be the set guaranteed by Theorem 4.28, let  $w \notin A$ , and let  $B = A \cup \{w\}$ . Suppose  $C$  is such that  $\|A\|_p \preceq \|C\|_p \preceq \|B\|_p$ , but  $C$  is not  $p$ -equipollent to  $B$ . We will show that  $C$  must be  $p$ -equipollent to  $A$ . Since  $\|C\|_p \preceq \|B\|_p$  by assumption, there is a  $p$ -equipollence  $f: C \rightarrow C'$  to a proper subset  $C' \subset B$ . Since neither the conclusion nor the statement depends on the representative of  $\|C\|_p$  chosen, without loss of generality let us replace  $C$  by  $C'$ .

Since  $B = A \cup \{w\}$  and  $C' \subseteq B$ ,  $C'$  is either a subset of  $A$ , or is  $A' \cup \{w\}$  for a subset  $A' \subseteq A$ . But by assumption, there is a  $p$ -equipollence from  $A$  to a subset of  $C'$ . By Theorem 4.28(1),  $C'$  cannot be a proper subset of  $A$ . If  $C' = A$ , then we are done.

So all that is left is to handle the case that  $C' = A' \cup \{w\}$  for some  $A' \subseteq A$ . If  $A' = A$ , then  $C' = B$ , contradicting our assumption that  $\|C\|_p \not\preceq \|B\|_p$ . Thus  $A'$  must be a proper subset of  $A$ . If  $|A \setminus A'| \geq 2$ , then let  $w_1 \in A \setminus A'$ . Then we have  $\|C'\|_p = \|A' \cup \{w\}\|_p = \|A' \cup \{w_1\}\|_p$ . But  $A' \cup \{w_1\}$  is a proper subset of  $A$ , contradicting Theorem 4.28(1). So  $|A \setminus A'| \leq 2$ , and since  $A'$  is a proper subset of  $A$ , we must have  $|A \setminus A'| = 1$ . Thus  $|C' \setminus A| - |A \setminus C'| = |\{w\}| - |A \setminus A'| = 1 - 1 = 0$ , so by Theorem 4.28(2),  $\|C'\|_p = \|A\|_p$ .  $\square$

## 5 Statements Weaker than the Axiom of Choice

Here we consider statements that are implied by the Axiom of Choice over ZF, but that are not known to be equivalent to AC (or are known to be strictly weaker), because their polynomial-time analogues make interesting connections.

**Classical Result 1** Any union of countably many countable sets is countable.

**Polynomial-Time Analogue 1** If  $L$  is a  $\mathsf{P}$  collection of languages  $L_x$  each of which is  $p$ -countable, then  $L$  is  $p$ -countable.

One might hope to build a  $p$ -equipollence  $L \rightarrow \Sigma^*$  by using the individual  $p$ -equipollences  $L_x \rightarrow \Sigma^*$  together with the fact that  $\|\Sigma^*\|_p = \|\Sigma^* \times \Sigma^*\|_p$ . However, one runs into the issue that if  $f_x: L_x \rightarrow \Sigma^*$  is a  $p$ -equipollence, there may not be a uniform polynomial upper bound on the runtimes of all the  $f_x$ . In the opposite direction, the fact that  $L$  itself is in  $\mathsf{P}$  means there *is* a uniform polynomial upper bound on the times to decide each  $L_x$ , so perhaps there is still some hope that this is unconditionally true. We leave it as an open question.

**Open Question 5.1** Prove or disprove the PTA1. Does one of the polynomial-time versions of AC imply PTA1?

**Classical Result 2** If  $A$  is infinite, there is an injection from  $\mathbb{N} \rightarrow A$ .

A natural  $p$ -analogue of this fails:

**Corollary 5.1** *There is an infinite language  $L \in \mathsf{P}$  such that  $\|\Sigma^*\|_p \not\preceq \|L\|_p$ .*

This is a corollary to the proof of Proposition 4.39.

*Proof* Let  $L$  be one of the languages constructed in Proposition 4.39.  $\|\Sigma^*\|_p \leq L$  would imply  $L$  is exponentially dense, but  $L$  is infinitely-often linearly sparse, so the result follows.  $\square$

**Classical Result 3** (The Axiom of Uniformization) If  $R \subset X \times Y$  where  $X$  and  $Y$  are Polish spaces, then there is a subset  $f \subseteq R$  that is a partial function  $f: X \rightarrow Y$  and such that  $\text{dom}(f) = \{x \in X \mid \exists y \in Y(x, y) \in R\}$ .

**Polynomial-Time Analogue 2** If  $X, Y \in \mathbf{P}$  and  $R \subseteq X \times Y$  is in  $\mathbf{P}$  as well, then there is a refinement  $f$  of  $R$  such that  $f$  is a partial function that is polynomial-time computable and whose domain is exactly  $\text{dom}(f) = \{x \in X : \exists y \in Y(x, y) \in R\}$ .

This is a restatement of  $\text{NPMV}_g \subseteq_c \text{PF}$ , studied by Blass & Gurevich [6] about a decade before Selman introduced the  $\text{NPMV}$  notation [63]. Note that some of our polynomial-time versions of AC are about  $\text{NPMV}_t$  or  $\text{NPMV}_{gt}$ , but collapses of  $\text{NPMV}_t$  are generally not known to imply collapses of  $\text{NPMV}$ , in contrast to the classical, unbounded situation where AC implies the Axiom of Uniformization.

## 6 Discussion and Future Work

Beyond the many questions and conjectures already stated (Questions 1.2, 3.7, 4.29, 4.34, 4.38, 4.47, 4.51, questions around Theorem 4.25, Conjectures 4.9 and 4.35), here we raise a few more that we think would be interesting avenues of exploration.

*Around the Isomorphism Conjecture.* Are Joseph & Young's creative sets [38], proposed as counterexamples to the Isomorphism Conjecture [5],  $p$ -equipollent to each other? To SAT? (See, e. g., the introduction of [42] for a survey of developments around this.) Recall that a language  $A$  is a *cylinder* if  $A \cong^p A \times \Sigma^*$ ; this is equivalent to  $A$  being *paddable*, and cylinders play an important role more generally in investigations around the Isomorphism Conjecture [5]. One of the main theorems of Berman & Hartmanis [5] is that an NP-complete set is  $p$ -isomorphic to SAT iff it is a cylinder. Note that if  $A$  is a cylinder then  $\|A\|_p = \|A \times \Sigma^*\|_p \geq \|\Sigma^*\|_p$ . Note that  $A$  is a cylinder iff  $\overline{A}$  is. What can be said about the cardinalities of cylinders? If  $A$  and  $\overline{A}$  both have  $p$ -cardinalities equal to their product with  $\Sigma^*$ , must  $A$  be cylinder-like in some way (cf. Question 4.34)?

*Connections to other complexity notions.* Can some of our open questions about  $p$ -cardinality or polynomial-time axioms of choice be resolved assuming  $\mu_p(\text{NP}) \neq 0$  [43]?

It seems there should be interesting connections to be had between  $p$ -cardinality and several of Selman's other interests, such as  $p$ -selectivity [34, 48, 57–59], disjoint pairs [17, 22–24], and mitoticity [18–20].

Several versions of AC use well-orderings. In the context of studying self-reducibility, Meyer & Paterson introduced some definitions that could be useful for polynomial-time analogues of these versions of AC:

**Definition 6.1** (Meyer & Paterson [46, Definition 4]) A partial order  $<$  on  $\Sigma^*$  is *polynomially well-founded* if there is a polynomial  $p$  such that every finite  $<$ -decreasing chain has at most  $p(|x|)$  elements in it, where  $|x|$  is the maximal length of any element in the chain.

A language  $L \subseteq \Sigma^*$  is *self-reducible* if there is a polynomial-time oracle TM  $M^\square$  and an  $p$ -well-founded, length-related ordering  $<$  such that  $L(M^L) = L$  and for any input  $x$ ,  $M^L(x)$  only queries the oracle on strings that are strictly  $< x$ .

Selman studied this notion of self-reducibility in (at least) [33, 59, 60].

**Classical AC 10** (See [53, WE1, p. 1]) Every set can be well-ordered.

**Polynomial-Time AC (PAC) 8** Every language in  $P$  admits a  $p$ -well-ordering computable in  $P$ .

More generally, aside from the use of well-orderings in relation to AC, an anonymous reviewer has suggested, and we agree, that it would be interesting to develop a theory of *p-ordinals*, similar to the development of  $p$ -cardinals here. We point out that this could either be in terms of total orders on  $\Sigma^*$  computable in FP, or in terms of total orders on subsets  $A \subseteq \Sigma^*$ , where the order relation is computable in PF on  $A \times A$ . It is unclear to us at this point which would fit more naturally with our development of  $p$ -cardinals here.

Recall that a set is said to be Dedekind finite iff it does not have a bijection onto any of its proper subsets. In classical set theory, “finite” and “Dedekind finite” are equivalent, but in other settings this need not be the case. Dekker [10] and Dekker–Myhill [11] were originating works on isols as models of Dedekind-finite sets.

**Definition 6.2** (Dedekind  $p$ -finite) A language  $L \subseteq \Sigma^*$  is *Dedekind  $p$ -finite* if it does not have the same  $p$ -cardinality as any of its proper subsets.

The set constructed in Theorem 4.28 is an infinite set that is Dedekind  $p$ -finite. How do Dedekind  $p$ -finite sets relate to the  $p$ -finite sets of Nerode & Rempel [51, Definition 6]? What can Dedekind  $p$ -finite sets tell us about  $p$ -cardinality more generally? Do they have a nice theory of arithmetic? Can they be used to realize Downey’s suggestion [12] (see the quote in Section 1)?

Lastly, one could also consider finite axioms of choice, in which one starts with a collection consisting of *finite* sets—here one may think correspondingly of a language  $L$ , where a single element of  $L$  represents a binary set by considering the binary encoding of  $x$  as the characteristic function of a finite subset of  $\mathbb{N}$ . Finite AC are discussed in [37, Section 2.2] and [66].

## Appendix A: Comments on Other Versions of the Axiom of Choice

In this appendix we gather some versions of AC we came across whose polynomial-time versions seem feasible and interesting to study, but we have not undertaken such explorations.

## A.1 Algebraic Versions

**Classical AC 11** Every vector space has a basis.

Nerode and Rempel [52] give oracles  $A_1, A_2$  relative to which  $P^{A_1} \neq NP^{A_1}$ , and relative to  $A_1$  every non-deterministic polynomial-time subspace has a polynomial-time basis, while relative to  $A_2$  there are non-deterministic polynomial-time subspaces with no polynomial-time bases. But, as with the rest of their line of work, here vectors are all encoded over a unary alphabet.

**Classical AC 12** For every non-empty set  $S$  there is a binary operation defined on  $S$  that makes  $S$  a group.

The polynomial-time version of this seems related to the notions of groupy witnesses [16, Section 4.1.2] and group definability [2].

**Classical AC 13** (Tychonoff's Theorem, see [53, P5, p. 69]) The product of compact spaces is compact.

## A.2 Model-Theoretic Versions

**Classical Result 4** Every game  $G_S$  in which  $S$  is a Borel subset of Baire space is determined.

**Classical AC 14** (See [53, P6, p. 69]) A formula having a model in a set of cardinality  $n$  also has a model in a set of cardinality  $m$  if  $\aleph_0 \leq m \leq n$ .

**Classical AC 15** (See [53, P7, p. 69]) A formula having a model in a set of cardinality  $\aleph_0$  also has a model in a set of any cardinality greater than  $\aleph_0$ .

**Classical AC 16** (See [53, P8, p. 69]) If  $Q$  is a set of formulas in which the set of individual constants has cardinality  $m$  and every finite subset of  $Q$  has a model, then  $Q$  has a model in a set whose cardinality is not greater than  $m + \aleph_0$ .

## A.3 Order-Theoretic Versions

**Classical AC 17** (Hausdorff Maximum Principle, see [37, Theorem 2.2]) In any poset, every totally ordered subset is contained in a maximal totally ordered subset. Equivalently: every poset (merely) has a maximal totally ordered subset.

**Classical AC 18** (Kurepa's Maximal Antichain Condition, see [37, Theorem 2.4(3)]) Every poset has a maximal antichain.

Recall that a collection  $\mathcal{S}$  of sets is *of finite character* if for every set  $A \in \mathcal{S}$ , every finite subset  $F \subseteq A$  is also in  $\mathcal{S}$ , and conversely, that is, if there exists a set  $A$  such

that every finite subset  $F \subseteq A$  is in  $\mathcal{S}$ , then  $A$  is also in  $\mathcal{S}$ .

**Classical AC 19** (Tukey’s Lemma, see [37, Theorem 2.2], [53, M7, p. 13]) Every non-empty collection of finite character has a maximal element with respect to inclusion.

**Polynomial-Time AC (PAC) 9** Every  $\mathcal{P}$  collection of honestly non-empty languages of finite character has a maximal element with respect to inclusion.

Finally, we mention a few based on cardinal exponentiation. Cardinal exponentiation is usually defined as  $|A|^{|B|} := |\{f : A \rightarrow B\}|$ .

**Classical AC 20** (See [53, CN23, CN24, CN25, pp. 53–54])

1. For each fixed infinite cardinal  $m$ , for all infinite cardinals  $p, q$ , if  $p^m < q^m$  then  $p < q$ .
2. There is a cardinal  $n > 1$  such that for all cardinals  $p, q$  there is a cardinal  $1 < m \leq n$  such that  $p^m < q^m$  implies  $p < q$ .
3. For all  $m, p, q$ , if  $m^p < m^q$  and  $m \neq 0$  then  $p < q$ .

On the one hand, when  $B$  is finite, we should have  $A^B = A \times A \times \dots \times A$ , where the latter product has  $|B|$  copies of  $A$ . On the other hand, when  $B$  is infinite, it feels natural to potentially define the  $p$ -exponent  $A^B$  as the set of Turing machines (/programs/indices) that compute partial polynomial-time functions  $B \rightarrow A$ . This causes two issues: first, when  $B$  is finite, these two definitions disagree. Second, using the latter definition, whenever  $A, B$  are nonempty,  $A^B$  is uncomputable by Rice’s Theorem. We nonetheless make the following provisional definition:

**Definition A.1** (Exponentiation of  $p$ -cardinals) Let  $A \subseteq \Sigma^*$ ,  $B \subseteq \Gamma^*$ . We define  $\|A\|_p^{\|B\|_p}$  as the  $p$ -cardinality of the set of Turing machines  $M$  such that  $M$  computes a polynomial-time partial function  $B \rightarrow A$ .

Let us unwrap what it means for expressions such as  $\|A\|_p^{\|C\|_p} \leq \|B\|_p^{\|C\|_p}$ . This means there is a partial polynomial-time computable,  $p$ -invertible function  $f$  (invertible on  $\text{img}(f)$ ) that, given any Turing machine  $M$  computing a polynomial-time function  $C \rightarrow A$ , outputs a Turing machine  $f(M)$  computing a partial polynomial-time function  $C \rightarrow B$ . This gives us some hope that, despite the uncomputability of  $\|A\|_p^{\|B\|_p}$ , meaningful statements might be provable about it.

**Polynomial-Time AC (PAC) 10**

1. For each fixed  $C \in \mathcal{P}$ , for all languages  $A, B \in \mathcal{P}$ , if  $\|A\|_p^{\|C\|_p} < \|B\|_p^{\|C\|_p}$ , then  $\|A\|_p < \|B\|_p$ .
2. There is an set  $N \in \mathcal{P}$  with  $\|N\|_p > 1$  such that for all sets  $A, B \in \mathcal{P}$ , there is a set  $1 < \|C\|_p \leq \|N\|_p$  in  $\mathcal{P}$  such that  $\|A\|_p^{\|C\|_p} < \|B\|_p^{\|C\|_p}$  implies  $\|A\|_p < \|B\|_p$ .
3. For all nonempty  $C \in \mathcal{P}$  and all languages  $A, B \in \mathcal{P}$ , if  $\|C\|_p^{\|A\|_p} < \|C\|_p^{\|B\|_p}$ , then  $\|A\|_p < \|B\|_p$ .

## A.4 Forms of AC which it was Difficult to Formulate an Interesting Polynomial-Time Analogue

Here we catalog some of the versions of the Axiom of Choice for which we had difficulty formulating a reasonable polynomial-time analogue, and the difficulties we encountered.

The first few make reference to the power set, which caused us trouble:

**Classical AC 21.** For any set  $A$ , its power set (with the empty set removed) has a choice function. Equivalently: for any set  $A$  there is a function  $f$  such that for any nonempty subset  $B \subseteq A$ ,  $f(B) \in B$ .

**Classical AC 22** (See [37, Theorem 2.4(5)]). The power set of each well-orderable set can be well-ordered.

On the one hand, we would like to talk about the “polynomial-time power set” of a language  $L \in \mathcal{P}$ . One attempt is to consider the collection of all  $L' \subseteq L$  such that  $L' \in \mathcal{P}$ ; a natural way to refer to these subsets  $L'$  is to write  $L' = L \cap L(M_x)$  for some polynomial-time machine  $M_x$ , and then these indices  $x$  can be used as the input to a choice function or well-ordering function. But then we run into the issue that any such polynomial-time function cannot even simulate all  $M_x$ , given  $x$  as input.

Others made reference to the existence of a single element, which felt not amenable to asymptotic considerations, e. g., perhaps one of the most famous:

**Classical AC 23** (Zorn’s Lemma, see, e. g., [37, Theorem 2.2]). Every non-empty poset in which every chain (totally ordered subset) has an upper bound contains at least one maximal element.

**Acknowledgements** We thank Lance Fortnow for initial discussions and some preliminary results on these problems when we first discussed them (I was in graduate school), and also for suggesting I submit it in memoriam. We thank two anonymous reviewers for their suggestions, which both improved the exposition and organization of the paper, and added results to it. In particular, Corollary 4.46 was originally posed as an open question, but because of expository improvements to the proof of Theorem 4.43 suggested by the reviewers, it needed only a 1-line modification to the original proof of that theorem (the theorem in the original submission did not guarantee that  $C$  was infinite). Theorem 4.28 answered an open question posed in the original submission, using a modification of a construction suggested by one of the reviewers.

The preparation of this work was partially supported by NSF CAREER award CCF-2047756, and probably one of Lance’s awards from when I was in graduate school, but that was long enough ago we don’t remember and the NSF probably no longer cares.

And of course we thank Alan for his immense curiosity and insight that he shared so well with us all.

**Open Access** This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article’s Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article’s Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

## References

1. Allender, E.A.: Invertible functions. PhD thesis, Georgia Institute of Technology, 1985. [https://people.cs.rutgers.edu/~allender/publications/complete\\_list.html](https://people.cs.rutgers.edu/~allender/publications/complete_list.html)



2. Arvind, V., Vinodchandran, N.V.: The counting complexity of group-definable languages. *Theoret. Comput. Sci.* **242**(1-2), 199–218 (2000). [https://doi.org/10.1016/S0304-3975\(98\)00218-7](https://doi.org/10.1016/S0304-3975(98)00218-7)
3. Ash, C.J., Knight, J.: *Computable Structures and the Hyperarithmetical Hierarchy*, Volume 144 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam (2000)
4. Baker, T.P., Selman, A.L.: A second step toward the polynomial hierarchy. *Theoret. Comput. Sci.* **8**(2), 177–187 (1979). [https://doi.org/10.1016/0304-3975\(79\)90043-4](https://doi.org/10.1016/0304-3975(79)90043-4)
5. Berman, L., Hartmanis, J.: On isomorphisms and density of NP and other complete sets. *SIAM J. Comput.* **6**(2), 305–322 (1977). <https://doi.org/10.1137/0206023>
6. Blass, A., Gurevich, Y.: Equivalence relations, invariants, and normal forms, II. In: *Logic and Machines: Decision Problems and Complexity*, volume 171 of *Lecture Notes in Computer Science*, pages 24–42. Springer (1984). [https://doi.org/10.1007/3-540-13331-3\\_31](https://doi.org/10.1007/3-540-13331-3_31)
7. Buss, S.R., Hay, L.: On truth-table reducibility to SAT. *Inform. and Comput.* **91**(1), 86–102 (1991). Preliminary version in *Structure in Complexity Conf. '88*. [https://doi.org/10.1016/0890-5401\(91\)90075-D](https://doi.org/10.1016/0890-5401(91)90075-D)
8. Carl, M.: Generalized effective reducibility. In: Beckmann, A., Bienvenu, L., Jonoska, N. (eds.) *Pursuit of the Universal*, pp. 225–233. Springer International Publishing, Cham (2016). arXiv version [1601.01899](https://arxiv.org/abs/1601.01899) [math.LO]. [https://doi.org/10.1007/978-3-319-40189-8\\_23](https://doi.org/10.1007/978-3-319-40189-8_23)
9. Csimá, B., Goncharov, S., Greenberg, N., Knight, J., Montalbán, A., Slaman, T.: *Computable model theory, report from a BIRS workshop*. <https://www.birs.ca/workshops/2013/13w5047/report13w5047.pdf>. Accessed 23 March 2023 (2014)
10. Dekker, J.C.E.: The recursive equivalence type of a class of sets. *Bull. Amer. Math. Soc.* **70**, 628–632 (1964). <https://doi.org/10.1090/S0002-9904-1964-11215-5>
11. Dekker, J.C.E., Myhill, J.: Recursive equivalence types. *Univ. California Publ. Math.* **3**, 67–213 (1960)
12. Downey, R.G.: Review of “Polynomially isolated sets” by A. Nerode and J. B. Remmel. *MathSciNet Review of MR1071525*. <https://mathscinet.ams.org/mathscinet-getitem?mr=1071525>. Accessed 23 March 2023 (1990)
13. Dzhafarov, D.D., Mummert, C.: Reverse mathematics and equivalents of the axiom of choice. arXiv:[1009.3242](https://arxiv.org/abs/1009.3242) [math.LO] (2010)
14. Fenner, S.A., Fortnow, L., Naik, A.V., Rogers, J.D.: Inverting onto functions. *Inform. and Comput.* **186**(1), 90–103 (2003). [https://doi.org/10.1016/S0890-5401\(03\)00119-6](https://doi.org/10.1016/S0890-5401(03)00119-6)
15. Fenner, S.A., Kurtz, S.A., Royer, J.S.: Every polynomial-time 1-degree collapses if and only if  $P = PSPACE$ . *J. Symbolic Logic* **69**(3), 713–741 (2004). <https://doi.org/10.2178/jsl/1096901763>
16. Fortnow, L., Grochow, J.A.: Complexity classes of equivalence problems revisited. *Inform. and Comput.* **209**(4), 748–763 (2011). Preprint [arXiv:0907.4775](https://arxiv.org/abs/0907.4775)[cs.CC]. <https://doi.org/10.1016/j.ic.2011.01.006>
17. Glaßer, C., Hughes, A., Selman, A.L., Wisiol, N.: Disjoint NP-pairs and propositional proof systems. *SIGACT News* **45**(4), 59–75 (2014). <https://doi.org/10.1145/2696081.2696095>
18. Glaßer, C., Nguyen, D.T., Selman, A.L., Witek, M.: Introduction to autoreducibility and mitoticity. In: Day, A.R., Fellows, M.R., Greenberg, N., Khoussainov, B., Melnikov, A.G., Rosamond, F.A. (eds.) *Computability and Complexity - Essays Dedicated to Rodney G. Downey on the Occasion of His 60th Birthday*, volume 10010 of *Lecture Notes in Computer Science*, pages 56–78. Springer (2017). [https://doi.org/10.1007/978-3-319-50062-1\\_5](https://doi.org/10.1007/978-3-319-50062-1_5)
19. Glaßer, C., Ogihara, M., Pavan, A., Selman, A.L., Zhang, L.: Autoreducibility, mitoticity, and immunity. *J. Comput. Syst. Sci.* **73**(5), 735–754 (2007). Preliminary version in *MFCS '05*. <https://doi.org/10.1016/j.jcss.2006.10.020>
20. Glaßer, C., Pavan, A., Selman, A.L., Zhang, L.: Mitosis in computational complexity. In: Cai, J.-Y., Cooper, S.B., Li, A. (eds.) *Theory and Applications of Models of Computation, Third International Conference, TAMC 2006, Beijing, China, May 15-20, 2006, Proceedings*, volume 3959 of *Lecture Notes in Computer Science*, pages 61–67. Springer (2006). [https://doi.org/10.1007/11750321\\_4](https://doi.org/10.1007/11750321_4)
21. Glaßer, C., Pavan, A., Selman, A.L., Zhang, L.: Splitting NP-complete sets. *SIAM J. Comput.* **37**(5), 1517–1535 (2008). <https://doi.org/10.1137/060673886>
22. Glaßer, C., Selman, A.L., Sengupta, S.: Reductions between disjoint NP-pairs. In: *19th Annual IEEE Conference on Computational Complexity (CCC 2004)*, 21-24 June 2004, Amherst, MA, USA, pages 42–53. IEEE Computer Society (2004). <https://doi.org/10.1109/CCC.2004.1313791>
23. Glaßer, C., Selman, A.L., Sengupta, S., Zhang, L.: Disjoint NP-pairs. *SIAM J. Comput.* **33**(6), 1369–1416 (2004). Preliminary version in *CCC '03*. <https://doi.org/10.1137/S0097539703425848>

24. Glaßer, C., Selman, A.L., Travers, S.D., Wagner, K.W.: The complexity of unions of disjoint sets. *J. Comput. Syst. Sci.* **74**(7), 1173–1187 (2008). <https://doi.org/10.1016/j.jcss.2008.05.001>
25. Goldberg, A.V., Sipser, M.: Compression and ranking. *SIAM J. Comput.* **20**(3), 524–536 (1991). <https://doi.org/10.1137/0220034>
26. Goldsmith, J., Homer, S.: Scalability and the isomorphism problem. *Inform. Process. Lett.* **57**(3), 137–143 (1996). [https://doi.org/10.1016/0020-0190\(95\)00204-9](https://doi.org/10.1016/0020-0190(95)00204-9)
27. Goldsmith, J., Kunen, K., Hemachandra, L.A.: Polynomial-time compression. *Comput. Complexity* **2**, 18–39 (1992). <https://doi.org/10.1007/BF01276437>
28. Grochow, J.A.: The complexity of equivalence relations. Master's thesis, University of Chicago. [https://home.cs.colorado.edu/~jgrochow/Grochow\\_UofC\\_Masters\\_08\\_Equivalence\\_Relations.pdf](https://home.cs.colorado.edu/~jgrochow/Grochow_UofC_Masters_08_Equivalence_Relations.pdf)
29. Grochow, J.A., Qiao, Y.: On p-Group Isomorphism: Search-to-decision, counting-to-decision, and Nilpotency class reductions via tensors. In: Kabanets, V. (ed.) 36th Computational Complexity Conference (CCC 2021), volume 200 of Leibniz International Proceedings in Informatics (LIPIcs), pp. 16:1–16:38. Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl, Germany (2021). <https://doi.org/10.4230/LIPIcs.CCC.2021.16>
30. Grollmann, J., Selman, A.L.: Complexity measures for public-key cryptosystems. *SIAM J. Comput.* **17**(2), 309–335 (1988). Special issue on cryptography. <https://doi.org/10.1137/0217018>
31. Hemachandra, L.A.: Counting in structural complexity theory. PhD thesis, Cornell University. L. A. Hemachandra is the pre-marriage name of L. A. Hemaspaandra (1987)
32. Hemachandra, L.A., Hoene, A., Siefkes, D., Young, P.: On sets polynomially enumerable by iteration. *Theoret. Comput. Sci.* **80**(2), 203–225 (1991). Symposium on Mathematical Foundations of Computer Science '89. [https://doi.org/10.1016/0304-3975\(91\)90388-1](https://doi.org/10.1016/0304-3975(91)90388-1)
33. Hemaspaandra, E., Naik, A.V., Ogihara, M., Selman, A.L.: P-selective sets and reducing search to decision vs self-reducibility. *J. Comput. System Sci.* **53**(2, part 1), 194–209 (1996). Eighth Annual Structure in Complexity Theory Conference (San Diego, CA, 1993). <https://doi.org/10.1006/jcss.1996.0061>
34. Hemaspaandra, L., Torenvliet, L.: Theory of semi-feasible algorithms. Monographs in Theoretical Computer Science. An EATCS Series. Springer. <https://doi.org/10.1007/978-3-662-05080-4> (2003)
35. Hemaspaandra, L.A., Naik, A.V., Ogihara, M., Selman, A.L.: Computing solutions uniquely collapses the polynomial hierarchy. *SIAM J. Comput.* **25**(4), 697–708 (1996). <https://doi.org/10.1137/S0097539794268315>
36. Hemaspaandra, L.A., Selman, A.L. (eds.): Complexity theory retrospective. II. Springer-Verlag, New York (1997). <https://link.springer.com/book/9780387949734>
37. Herrlich, H.: Axiom of choice, volume 1876 of Lecture Notes in Mathematics. Springer-Verlag, Berlin (2006). <https://doi.org/10.1007/11601562>
38. Joseph, D., Young, P.: Some remarks on witness functions for nonpolynomial and noncomplete sets in NP. *Theoret. Comput. Sci.* **39**(2-3), 225–237 (1985). [https://doi.org/10.1016/0304-3975\(85\)90140-9](https://doi.org/10.1016/0304-3975(85)90140-9)
39. Ko, K.-I.: On some natural complete operators. *Theoret. Comput. Sci.* **37**(1), 1–30 (1985). [https://doi.org/10.1016/0304-3975\(85\)90085-4](https://doi.org/10.1016/0304-3975(85)90085-4)
40. Ko, K.-I., Moore, D.: Completeness, approximation and density. *SIAM J. Comput.* **10**(4), 787–796 (1981). <https://doi.org/10.1137/0210061>
41. Kurtz, S.A., Mahaney, S.R., Royer, J.S.: Collapsing degrees. *J. Comput. Syst. Sci.* **37**, 247–268 (1988). [https://doi.org/10.1016/0022-0000\(88\)90007-4](https://doi.org/10.1016/0022-0000(88)90007-4)
42. Kurtz, S.A., Mahaney, S.R., Royer, J.S.: The isomorphism conjecture fails relative to a random oracle. *J. Assoc. Comput. Mach.* **42**(2), 401–420 (1995). <https://doi.org/10.1145/201019.201030>
43. Lutz, J.H.: Category and measure in complexity classes. *SIAM J. Comput.* **19**(6), 1100–1131 (1990). <https://doi.org/10.1137/0219076>
44. McCarty, C.: Realizability and recursive mathematics. PhD thesis, University of Oxford. <http://hdl.handle.net/1842/35199> (1985)
45. McCarty, C.: Realizability and recursive set theory. *Ann. Pure Appl. Logic* **32**(2), 153–183 (1986). [https://doi.org/10.1016/0168-0072\(86\)90050-3](https://doi.org/10.1016/0168-0072(86)90050-3)
46. Meyer, A.R., Paterson, M.S.: With what frequency are apparently intractable problems difficult? MIT Tech. Report MIT/LCS/TM-126. <http://publications.csail.mit.edu/lcs/pubs/pdf/MIT-LCS-TM-126.pdf> (1979)
47. Montalbán, A.: Computable structure theory. Perspectives in Logic. Cambridge University Press, Cambridge; Association for Symbolic Logic, Ithaca, NY. Within the arithmetic. <https://doi.org/10.1017/9781108525749> (2021)

48. Naik, A.V., Selman, A.L.: A note on p-selective sets and on adaptive versus nonadaptive queries to NP. In: Homer, S., Cai, J.-Y. (eds.) Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity, Philadelphia, Pennsylvania, USA, May 24–27, 1996, pages 224–232. IEEE Computer Society (1996). <https://doi.org/10.1109/CCC.1996.507684>
49. Nerode, A., Rempel, J.B.: Complexity-theoretic algebra. II. Boolean algebras. *Ann. Pure Appl. Logic* **44**(1–2), 71–99 (1989). Third Asian Conference on Mathematical Logic (Beijing, 1987). [https://doi.org/10.1016/0168-0072\(89\)90047-X](https://doi.org/10.1016/0168-0072(89)90047-X)
50. Nerode, A., Rempel, J.B.: Polynomial time equivalence types. In: Logic and computation (Pittsburgh, PA, 1987), volume 106 of *Contemp. Math.*, pages 221–249. Amer. Math. Soc., Providence, RI (1990). <https://doi.org/10.1090/conm/106/1057825>
51. Nerode, A., Rempel, J.B.: Polynomially isolated sets. In: Recursion theory week (Oberwolfach, 1989), volume 1432 of *Lecture Notes in Math.*, pages 323–362. Springer, Berlin (1990). <https://doi.org/10.1007/BFb0086125>
52. Nerode, A., Rempel, J.B.: Complexity theoretic algebra. I. Vector spaces over finite fields. Conference on Logic and Computer Science: New Trends and Applications (Turin, 1986). *Rend. Sem. Mat. Univ. Politec. Torino* 1987, Special Issue, 1–10 (1988)
53. Rubin, H., Rubin, J.E.: *Equivalents of the Axiom of Choice*. North-Holland Publishing Co., Amsterdam (1963)
54. Rubin, H., Rubin, J.E.: *Equivalents of the axiom of choice. II*, volume 116 of *Studies in Logic and the Foundations of Mathematics*. North-Holland Publishing Co., Amsterdam (1985)
55. Schechter, E.: *Handbook of analysis and its foundations*. Academic Press, Inc., San Diego (1997)
56. Selman, A.L.: Polynomial time enumeration reducibility. *SIAM J. Comput.* **7**(4), 440–457 (1978). <https://doi.org/10.1137/0207035>
57. Selman, A.L.: P-selective sets, tally languages, and the behavior of polynomial time reducibilities on NP. *Math. Syst. Theory* **13**, 55–65 (1979). <https://doi.org/10.1007/BF01744288>
58. Selman, A.L.: Some observations on NP, real numbers and p-selective sets. *J. Comput. Syst. Sci.* **23**(3), 326–332 (1981). [https://doi.org/10.1016/0022-0000\(81\)90068-4](https://doi.org/10.1016/0022-0000(81)90068-4)
59. Selman, A.L.: Reductions on NP and p-selective sets. *Theoret. Comput. Sci.* **19**(3), 287–304 (1982). [https://doi.org/10.1016/0304-3975\(82\)90039-1](https://doi.org/10.1016/0304-3975(82)90039-1)
60. Selman, A.L.: Natural self-reducible sets. *SIAM J. Comput.* **17**(5), 989–996 (1988). <https://doi.org/10.1137/0217062>
61. Selman, A.L.: Complexity theory retrospective. Springer-Verlag, New York (1990). In honor of Juris Hartmanis on the occasion of his sixtieth birthday. <https://doi.org/10.1007/978-1-4612-4478-3>
62. Selman, A.L.: Complexity classes for partial functions. In: Rozenberg, G., Salomaa, A. (eds.) *Current Trends in Theoretical Computer Science - Essays and Tutorials*, volume 40 of *World Scientific Series in Computer Science*, pages 504–522. World Scientific (1993). [https://doi.org/10.1142/9789812794499\\_0038](https://doi.org/10.1142/9789812794499_0038)
63. Selman, A.L.: A taxonomy of complexity classes of functions. *J. Comput. System Sci.* **48**(2), 357–381 (1994). [https://doi.org/10.1016/S0022-0000\(05\)80009-1](https://doi.org/10.1016/S0022-0000(05)80009-1)
64. Selman, A.L.: Much ado about functions. In: Homer, S., Cai, J.-Y. (eds.) Proceedings of the Eleventh Annual IEEE Conference on Computational Complexity, Philadelphia, Pennsylvania, USA, May 24–27, 1996, pages 198–212. IEEE Computer Society (1996). <https://doi.org/10.1109/CCC.1996.507682>
65. Tarski, A.: Sur quelques theorems qui equivalent a l'axiome du choix. *Fundam. Math.* **5**, 147–154 (1924). <http://matwbn.icm.edu.pl/ksiazki/fm/fm5/fm5118.pdf>
66. Truss, J.: Finite axioms of choice. *Ann. Math. Logic* **6**, 147–176 (1973/74). [https://doi.org/10.1016/0003-4843\(73\)90007-7](https://doi.org/10.1016/0003-4843(73)90007-7)
67. Wagner, K.W.: Log query classes. Technical Report TR-145, Universität Augsburg, Institut für Mathematik (1987)