The Reverse Mathematics of Hindman's Theorem

Jordan Brown

April 2020

Contents

1 Introduction

The field of reverse mathematics is devoted to ascertaining the proof-theoretic strengths of various results when they are assumed as axioms. In particular, given a fixed language, we can compare axiomatic systems (that is, sets of statements) $\mathcal{A}$ and $\mathcal{A}'$ in that language by asking whether every sentence in $\mathcal{A}'$ can be proven in $\mathcal{A}$ and vice versa. One may also ask whether $\mathcal{A} \cup \mathcal{A}'$ is a consistent set of sentences. In the past century, these questions have primarily been asked about set theory. One of the great achievements of twentieth-century logic was the demonstration that the axiom of choice is independent of the other axioms of Zermelo-Fraenkel set theory; that is, appending either it or its negation to the other axioms of ZF yields a consistent axiomatic system. It is also well-known that the axiom of choice is equivalent to several other results, such as the well-ordering theorem, over ZF. This simply means that, if we take the axioms of ZF and the axiom of choice as axioms, we can prove the well-ordering theorem, and if we take ZF and the well-ordering theorem as axioms we can prove the axiom of choice.

Here we examine such questions in a different language, the language of second-order arithmetic. Though not really a second-order language, the language of second-order arithmetic has two types of variables, numbers and sets, with relations corresponding to order, addition, and multiplication. It turns out that much of mathematics can be encoded in the language of second-order arithmetic, and given any theorem or set of theorems provable in the full axiomatic system of second-order arithmetic, we can consider the axiomatic system consisting of those statements and ask what can be proved from it.

Remarkably, most results fall into one of a few distinct classes of axiomatic strength. Subsystems of the axiomatic system of second-order arithmetic are considered to be the same if all of the axioms of one can be proved in the other and vice versa. Quite a few results are provable in a base system called RCA_0 corresponding to 'computable' mathematics, and those that are not usually induce one of four other 'Big Five' subsystems subsystems of the full axiomatic system of second-order arithmetic.

Hindman's theorem is a well-known combinatorial result, and like many known characterizations of Big Five subsystems, it is phrased as a statement about the existence of a set of natural numbers. It is known that Hindman's theorem lies near one of the Big Five systems, ACA_0, in terms of proof-theoretic strength, but it is not known whether it is equivalent to ACA_0. This is the primary open question about the proof-theoretic strength of Hindman's theorem, though there has also been significant recent work on the proof-theoretic strength of certain restrictions of Hindman's theorem by Carlucci.

There are close relationships between the hierarchy of subsystems of second-order arithmetic and the order of the Turing degrees, which we introduce below. From the perspective of computability theory, the status of Hindman's theorem is quite uncertain; in terms of reverse mathematics, the position of Hindman's theorem has very nearly been ascertained.

2 Second-Order Arithmetic

The language of second-order arithmetic, often denoted L_2, is a first-order, two-sorted language with a capital type and a miniscule type. It contains constant symbols 0 and 1 of miniscule type, binary function symbols $+$ and $\cdot$ that operate on letters of miniscule type and output a letter of miniscule type, a binary relation symbol $<$ between letters of miniscule type, and a binary relation symbol $\mathfrak{e}$ with first argument of miniscule type and second argument of capital type. This completes the formal description of the language. We will follow the custom of writing $+(a,b)$ as $a+b$, $\cdot(a,b)$ as ab, and $<(a,b)$ as $a<b$. The capital letters are to be interpreted as sets of natural number and the miniscule letters are to be interpreted as numbers, with $\mathfrak{e}$ representing set membership. We will usually refer to capital letters as set variables and miniscule letters as number variables. It should be stressed that we are working with full first-order logic when reasoning with L_2; there is no reason why we must adopt a constructive or otherwise restricted logical standpoint when studying arithmetic.

A (well-formed) formula in L_2 is Σ_0^0 if it contains no set quantifiers, the scope of any universal number quantifier $\forall n$ is of the form $(n \leq m \implies Q)$, and the scope of every existential number quantifier is of the form $(n \leq m \wedge Q)$, where Q is some well-formed formula; such quantifiers are said to be bounded. A formula is Π_0^0 if it is Σ_0^0. For $k>0$, a formula is Σ_k^0 if it is of the form $\exists n(\sim \Theta)$, where n is a miniscule letter variable and Θ is a Σ_{k-1}^0 formula. A formula is Π_k^0 if it is of the form $\forall n(\sim \Theta)$, where n is a miniscule letter variable and Θ is a Π_{k-1}^0 formula. A formula is Δ_k^0 if it is equivalent to both a Σ_k^0 formula and a Π_k^0 formula. The following (meta-mathematical) lemma is easily seen by induction.

Lemma 1. *For any $k \in \omega$, a formula Θ is Σ_k^0 if and only if $\sim \Theta$ is Π_k^0.*

The statement that a number is prime is Δ_0^0, for we can write it as $\varphi(n) = \forall j(j<n \implies (\forall k(k<n \implies (\sim (jk=n)))))$. Notationally, we will abbreviate $\forall n(n<m \implies Q)$ to $\forall (n<m)Q$ and similarly for bounded existential quantifiers, in which case the above becomes $(\forall j<n)(\forall k<n)(\sim (jk=n))$. We will also use symbols such as $\leq$ and $\neq$ with their usual meaning.

The axioms of RCA_0 are the following statements in L_2, where φ is a Σ_1^0 formula and ψ is a Δ_1^0 formula in which X does not occur:

1. $\forall n(n+0=n)$
2. $\forall n(0n=0)$
3. $\forall n(\sim (n<0))$
4. $\forall n(\sim (n+1=0))$
5. $\forall n \forall m(n+1=m+1 \implies n=m)$
6. $\forall n \forall m(m+(n+1)=(m+n)+1)$
7. $\forall n \forall m(n(m+1)=nm+n)$
8. $\forall n \forall m(n<m+1 \iff (n=m \vee n<m))$
9. $\forall m \exists n((\sim \varphi(0)) \vee (\varphi(n) \wedge (\sim \varphi(n+1)) \vee \varphi(m))$
10. $\exists X \forall n(n \mathfrak{e} X \iff \psi(n))$

If we had let φ, ψ be arbitrary formulae in L_2 (maintaining the restriction that X does not appear in ψ), we would have instead obtained Z_2, the full axiomatic system of second-order arithmetic. If we had required only that φ, ψ contain no set quantifiers (such formulae are said to be arithmetical) we would have obtained ACA_0. A subsystem of Z_2 is any collection of statements in L_2 that are provable in Z_2. An $\omega-$model of any such subsystem is a model where the universe of number variables is the usual set of natural numbers ω, the universe of set variables is some subset of $\mathcal{P}(\omega)$, and $\mathfrak{e}$ is interpreted as the membership relation $\in$.

Before proceeding further, let us see why these three systems are important. It turns out that RCA_0 is the weakest system in which a substantial body of mathematics can be developed. This is essentially because it allows us to form sets for which the criteria for membership can be computably determined. When the real numbers are appropriately defined in L_2, we are able to prove the Baire category theorem, the intermediate value theorem, and the existence of an algebraic closure for every countable field. Simpson proves these results in *Subsystems of Second-Order Arithmetic*.

Perhaps the best argument for why RCA_0 should be used as the base axiomatic system is that its minimal $\omega-$model contains precisely the computable sets. Intuitively, bounded number quantifiers can be computably verified, and Σ_1^0 sets are 'computably

enumerable' because membership in such a set can be decided by running a possibly nonterminating search for a number which satisfies a computable condition. By the above lemma it follows that a Δ^0_1 set is computably enumerable and its complement is computably enumerable, so it is computable (we will discuss computability more in the next section; for now simply think of a set as computable if there is an algorithm for determining if a number we are given is in the set). This is all formally and systematically developed in Simpson. (One should note that axiom 9, the induction axiom, is irrelevant in $\omega-$models, where induction automatically holds for all formulae, and thus the above argument only justifies treating the comprehension axiom 10 as a computable condition; axioms 1-8 must certainly be satisfied by anything called arithmetic.) The formal connection arises from the fact that the formation rules for partial recursive functions can all be expressed in L_2 and the existence of the resulting functions can be proved from RCA_0.

The system ACA_0 is almost as natural. It is a conservative extension of Peano (first-order) arithmetic; indeed, its first-order part, concerning statements only about numbers, is the same as that of Peano arithmetic. Like RCA_0, ACA_0 can be equivalently described by many statements using RCA_0 as a base system. By our above argument, unbounded existential quantifiers correspond to computably enumerable sets. Repeatedly applying this, we get that every set definable via an arithmetical formula is computable in some finite iterate of the Turing jump. Indeed we find that the minimal $\omega-$model of ACA_0 consists of sets computable with oracle $\emptyset^{(n)}$ for some $n < \omega$.

We will concern ourselves only with subsystems of Z_2 at least as strong as RCA_0. In terms of $\omega-$models, this means that we will only address questions regarding the inclusion of certain sets in the model, and as our prior statements about the connection between the comprehension axiom of RCA_0 and computability relativize, whenever an $\omega-$model of RCA_0 (and a fortiori any system at least as strong) contains some $X \subset \omega$, if Y is computable from X we must also have Y in the model. (Consider an appropriate Δ^0_1 formula with X as a free variable and Y as the set thus defined.) We reiterate that all of these statements can be precisely justified, and they are in Simpson. Because we only wish to consider subsystems in which RCA_0 holds, we will write HT for the axiomatic subsystem of Z_2 consisting of RCA_0 and Hindman's theorem.

Naturally, the above connections with computability theory are very powerful, and they are essential to our study of $\omega-$models of subsystems of Z_2. But they make little sense in other models. However, we can approximate applications of computability with formulae in L_2 using universal lightface Π^0_1 formulae. We will not examine the existence or properties of such formulae, but they are essential in generalizing results proven about $\omega-$models to all models of a subsystem of Z_2. In particular, they are used to derive, from the results of Blass, Hirst, and Simpson, the statements that HT proves ACA_0 and that HT is provable in ACA^+_0. In fact, they allow us to enumerate 'recursive' functions like we do with universal Turing machines, and this means that, if we have a statement involving quantification over all recursive functions, we can rewrite the statement using our indexing of recursive functions so that the quantification is of a number variable. This allows us to apply induction if we are assuming ACA_0.[1] When we embark on the main part of the argument, we will repeatedly implicitly use this. If you feel uncomfortable with discussing recursive functions (with or without oracles) in a setting that is not the 'true' natural numbers, you can either think of the argument as applying only to ω-models of second-order arithmetic or as applying over ACA, a subsystem of Z_2 that allows for arithmetical comprehension and arbitrary induction, as either will allow results proved by induction to go through when there are set quantifiers involved. But the argument is legitimate in the general case.

Though we will concern ourselves primarily with results that can be clearly formulated in L_2, we should note that, when definitions are made correctly, large bodies of mathematics can be expressed within L_2. Essentially any structure that can be 'countably controlled', such as a separable metric space, can be interpreted in L_2 in such a way that standard theorems can be proven in Z_2. But the machinery necessary to describe such applications is complex and centers around some technical arguments about encoding tuples and sets of natural natural numbers as natural numbers in a way amenable to use in RCA_0, so we will exclude such applications of second-order arithmetic. It is worth noting, however, that the structure of subsystems of Z_2 is often most easily seen through expressions as results in topology or analysis, and one of the major reasons why it is difficult to place Hindman's theorem and other Ramsey-type theorems within the hierarchy is that there is no known equivalent expression in terms of continuous mathematics.

Without going through the technical arguments mentioned above, we should briefly address pairing maps, as they are relevant to the statement of Hindman's theorem in L_2. If one wants to discuss pairs or tuples of natural numbers in L_2, one must define them to be abbreviations for something that is expressible in L_2, a natural number or a set of natural numbers. We call a 'function' that tells us how such tuples correspond to natural numbers a pairing map. There are many pairing maps if we do not require the pairing map to be surjective. Simpson uses (n, m) as an abbreviation for $(n + m)^2 + n$. (It is not really a 'map' in the language of second-order arithmetic, as it is not a number of a set of natural numbers!) Proceeding in this way we can

[1] The Blass, Hirst, Simpson paper is quite light on details, but this is the mechanism they are fundamentally relying on to push the argument through in ACA_0 instead of ACA.

encode arbitrarily many copies of $\mathbf{N}$ in $\mathbf{N}$ by repeatedly pairing, so that (a, b, c) is encoded as the natural number $((a, b), c)$. Thus we can consider functions from $\mathbf{N}$ to $\mathbf{N}$ as subsets of $\mathbf{N}$ itself once we have fixed a pairing map. We should also note that one may encode arbitrary finite sets as natural numbers; again, there are many ways to do this, but the most natural is easily seen to be the map that takes a finite set S to $\sum_{i \in S} 2^i$. This is not the encoding that we take in the development of second-order arithmetic, as we do not have the notion of such a sum before defining an encoding of finite sets as natural numbers, but the idea is fundamentally the same. We should remark here also that the term 'finite' refers to a formula expressible in L_2, not an external set-theoretic notion: we say that a set S is bounded if and only if there is some n such that $m \in S \implies m < n$. More concisely, $\exists n \forall m[m \in S \implies m < n]$. (Recall that we do not need to explicitly restrict the quantifiers to natural numbers, as this is exactly what we mean when we quantify a miniscule letter in L_2.) When we refer to a finite sum of elements in our discussion of Hindman's theorem, we are referring to a finite set in this internal, arithmetic sense.

3 Remarks on Computability

The definability of certain sets within subsystems of Z_2 is closely connected to their computablity-theoretic properties. So it is worthwhile giving a brief overview of the theory of computability here. The account is drawn primarily from Davis, which is generally an excellent reference for computability theory. Other good accounts can also be found in Weber and Rogers. Note that, in this section, we are only discussing the usual natural number system ω.

The formal model of computation we use is that of the Turing machine (or, rather, a formalization of Turing machines within set theory). Given a fixed, countable collections of distinct sets $q_0, q_1, ..., S_0, S_1, ...$ and some distinct sets L and R (which may be selected c whichever way one chooses), an instruction set is a finite set of ordered quadruples where the first component is some q_i, the second component is some S_i, the third component is either some q_k, S_j, L, or R, and the fourth component is some q_k, where no two elements of the instruction set have the same first and second components. For example, an instruction set could consist of the tuples $(q_2, S_0, S_3, q_5), (q_6, S_1, L, q_6), (q_8, S_2, q_4, q_5)$, and (q_2, S_2, S_4, q_3). A Turing machine with an oracle is an ordered pair $(\mathcal{S}, M)$, where $\mathcal{S}$ is an instruction set and $M \subset \mathbf{N}$. A Turing machine with oracle μ is a Turing machine with an oracle with second component equal to μ. A Turing machine is a Turing machine with oracle $\emptyset$. A simple Turing machine with an oracle is a Turing machine with an oracle which contains no quadruple where the third component is a state q_i. This completes the formal description of Turing machines.

The idea is that a Turing machine with an oracle acts on some infinite tape (with squares indexed by the integers) on which some symbols (S_i's) are printed by having a read/write head which scans one square on the tape at a time. If it is in state q_i and is reading symbol S_j and there is some quadruple in the machine with first two components being q_i and S_j, that 'instruction' tells the machine what to do: it either changes which square is scanned or it replaces the symbol currently in the square with another symbol. A total state is an ordered triple (T, s, r), where T is a function from $\mathbf{Z}$ to the set of symbols $S_0, S_1, ...$, s is one of the states $q_0, q_1, ...$, and r is an integer. We interpret T to indicate the state of the tape at a given time, s the state of the machine, and r the square that the machine is scanning.

Given a Turing machine with an oracle $(\mathcal{S}, M)$, we define a relation $\to$ (which would perhaps be more accurately denoted $\to_{(\mathcal{S},M)}$) on the set of total states by saying $(T, s, r) \to (T', s', r')$ if and only if one of the following hold:

1. $\mathcal{S}$ contains a quadruple of the form $(s, T(r), L, q_k)$ and $T' = T, s' = q_k, r' = r - 1$
2. $\mathcal{S}$ contains a quadruple of the form $(s, T(r), R, q_k)$ and $T' = T, s' = q_k, r' = r + 1$
3. $\mathcal{S}$ contains a quadruple of the form $(s, T(r), S_j, q_k)$ and $T'(n) = T(n)$ if $n \neq r$, $T'(r) = S_j$,$s' = q_k$, and $r' = r$
4. $\mathcal{S}$ contains a quadruple of the form $(s, T(r), q_j, q_k)$, the cardinality of the set $T^{-1}(S_1)$ is in M, $T' = T, s' = q_j, r' = r$
5. $\mathcal{S}$ contains a quadruple of the form $(s, T(r), q_j, q_k)$, the cardinality of the set $T^{-1}(S_1)$ is finite but not in M, $T' = T, s' = q_k, r' = r$

Because of our stipulation that an instruction set cannot have distinct elements that agree in both the first and second components, it follows that, for any total state (T, s, r), there is at most one total state (T', s', r') such that $(T, s, r) \to (T', s', r')$. A computation is a (finite or infinite) sequence γ of total states such that, for all natural numbers n, $\gamma(n) \to \gamma(n + 1)$, where γ is either defined on all of $\mathbf{N}$ or, if m is the greatest natural number on which it is defined, there is no total state (T, s, r) such that $\gamma(m) \to (T, s, r)$. In the latter case we say that the computation terminates or halts, and the output (or result) on input $\gamma(0)$ is T. In the former case we say that the machine gives no output on input $\gamma(0)$. For each total state, there is a unique computation γ such that $\gamma(0)$ is equal to that total state (if one assumes that there is no terminating computation beginning with that total state, one may recursively define an infinite sequence that forms a computation; uniqueness follows immediately from our prior

remarks).

If we interpret an input (initial total state) (T, s, r) where $s = q_0$, $r = 0$, and there is some natural n such that $T(x) = S_1$ if $0 \leq x < n$ and $T(x) = S_0$ otherwise as the natural number n, we can interpret the machine with an oracle as a partially defined function on the natural numbers, where if the machine terminates the output is the number of S_1's that appear in the final state and if the machine does not terminate the function is not defined on that input. We can also allow machines to act on tuples of natural numbers by encoding a tuple as a string of strings of S_1's with appearances of S_0 separating them, where the length of each string of S_1's encodes a natural number. Note that, if $M = \emptyset$, we can rewrite any machine with oracle M as a simple machine by just replacing appearances of (q_k, S_j, q_l, q_m), with (q_k, S_j, S_j, q_m). We say that a partially defined function on the natural numbers (or on some finite product of copies of the natural numbers) is $M-$computable or computable in M if there is a Turing machine with oracle M that returns no output precisely when the original function is not defined and, where the original function is defined, the Turing machine outputs the appropriate value. A function is said to be computable if it is computable in $\emptyset$ or, equivalently, if it can be computed by a simple Turing machine. A set of natural numbers is said to be $M-$computable if its characteristic function is $M-$computable.

We will hereafter ignore the above formalism. This is because of the Church-Turing thesis, which states that any function computable by an algorithm is computable. So if we describe a precise algorithm in words, we assume that it could be translated to the language of Turing machines. This is acceptable because, in each case where people have attempted to formalize an algorithm as a Turing machine, they have succeeded, and moreover every reasonable attempt at defining computable functions is equivalent to the Turing machine construction. The number of alternate formalizations is great, and for an introduction to many of them one should see Weber.

We chose the Turing machine formalism here because it has the clearest intuitive connection to computability; the image of a head following instructions based on symbols it reads is very compelling. However, to prove that all computable functions are definable in $\omega-$models of RCA_0, one does not use this characterization, but rather a definition that gives the set of functions computable from an oracle S as the smallest set of functions from the finite products of the natural numbers to natural numbers that includes constant maps, the successor function, and the characteristic function of S, and which is closed under operations corresponding to 'recursion' (defining the action on $n + 1$ in some clear way from the action on n). But the chosen operations we require closure over appear to be arbitrary, which is why the Turing machine model seems more natural.

It is generally accepted that every algorithm can be written as a Turing machine. Similarly, though the ways in which we can use oracles originally appears very limited, it essentially captures the entire notion of being able to refer to externally provided values.

If the oracle is itself computable, then it does nothing, as instead of referring to the oracle the machine could simply run through the computation to see if the number of S_1's was in M and proceed accordingly. It is easy to see that every set M is $M-$computable. In fact it is not difficult to see that this induces a pre-order on the set of sets of natural numbers by declaring $N \leq_T M$ if N is $M-$computable, and we will repeatedly use the transitivity of this order below without reference. (But actually writing out Turing machines for all of these things is quite tedious; see Davis for explicit constructions.) The equivalence classes over this pre-order are called Turing degrees. There are many open questions about the order structure; for instance, it is not known whether the set of Turing degrees admits a nontrivial automorphism. For more information about these open questions see Montalbán. Many mathematicians will naturally feel uncomfortable with proofs that rely on the Church-Turing thesis, but it is really no different from the ways in which we usually do mathematics. Papers are not written in formal languages; they indicate, in words, enough information so that the informed reader could translate them into the relevant formal proof structure without difficulty.

For any fixed oracle M, we can effectively enumerate all Turing machines with oracle M. That is, we can list the programs $\varphi_0^M, \varphi_1^M, ...$ with oracle M in such a fashion that one can write a universal Turing machine U^M such that, for all natural numbers e and n, $U^M(e, n) = \varphi_e^M(n)$ (where we interpret equality to include the statement that, wherever one side is defined, the other is defined as well). We will omit the details of this construction, but it boils down to fixing an encoding between natural numbers and programs that can be computed algorithmically; that is, at the beginning, U decodes e into a set of instructions, and then it proceeds to act on n where, at each step, the machine checks the set of instructions it determined from e at the beginning to determine how it should act before returning to the relevant square and proceeding accordingly. The existence of such a U gives us an effective listing of all programs with oracle M. We let M' be the Turing jump of M; that is, $M' \subset \mathbf{N}$ is given by $e \in M'$ if and only if φ_e^M halts on input e. One may naturally ask whether M' depends on the way in which we listed the machines with oracle M. Of course there is no canonical way to encode instructions as natural numbers, so there is no canonical way to effectively list the Turing machines. Hence we have not defined the set M' canonically, only with respect to the effective listing we fixed. However we have no issue with writing M' because, if we used some other listing and obtained some other jump

N, we would have that N is M'-computable and M' is $N-$computable. We will never be interested in the actual set M', only its computability properties, so one may choose whatever M' one likes. It is always true that M is M'-computable, but M' is never $M-$computable. Halting sets are very central in computability theory and were one of the primary objects studied in the classical theory of computability developed by Turing, Post, Kleene, and Church. Note that the above implies that there is no maximal Turing degree with respect to the order defined above. Of course the Turing jump can be iterated: the second jump of a set $X \subset \mathbf{N}$ is $(X')'$ and so on. We denote the n-th jump of X by $X^{(n)}$.

Any two Turing degrees have a least common upper bound, or join, with respect to the Turing order: given $X, Y \subset \mathbf{N}$, the join $X \oplus Y$ consists of even numbers n such that $n/2 \in X$ and odd numbers m such that $(m-1)/2 \in Y$. We leave it as an exercise to the reader to see that this operation is well-defined (choosing different representatives from the Turing degrees of X and Y give elements of the same Turing degree as the join) and it is a least upper bound for $\{X, Y\}$.

The Turing order captures the intuitive notion of a problem being 'even more noncomputable' than another. For example, the word problem for finitely presented groups is even more noncomputable than the halting problem: neither can be computed, but given an oracle for the word problem (when appropriately encoded in the natural numbers), one can compute the halting problem, though the converse is not the case. One may also think of adjoining algebraic elements to fields: there is no square root or fourth root of 2 in $\mathbf{Q}$, but there is a square root of 2 in $\mathbf{Q}[\sqrt[4]{2}]$ while there is no fourth root of 2 in $\mathbf{Q}[\sqrt{2}]$.[2]

One could consider allowing oracles that are subsets of any finite product of copies of $\mathbf{N}$, but this would not add anything because we have pairing maps that allow us to computably and injectively (in some increasing manner) embed products of $\mathbf{N}$ in $\mathbf{N}$ itself. (This is also how we can encode complex information about ordered pairs when working in second-order arithmetic, and is thus essential for defining e.g. the real numbers.)

On a final, notational note, we should say that the term 'recursive' is often used as a synonym for 'computable' both in this thesis and in the literature, and 'f is recursive in Y' means 'f is computable in Y' or, equivalently, 'f is Y-computable'.

4 König's Lemma

We devote this short section to a discussion of König's lemma and weak König's lemma, as they are important both to our proof of Hindman's theorem and the general structure of subsystems of second-order arithmetic. Assume we have fixed some encoding of finite sequences of natural numbers as natural numbers in RCA_0; we assume that finite sequences have some (possibly empty) initial segment $\{0, 1, ..., k-1, k\}$ of $\mathbf{N}$ as their domain. Then we say that a tree is a set T of such finite sequences where, if $f \in T$ and g is some finite sequence such that the domain of g is contained in the domain of f and, for all k in the domain of g, $g(n) = f(n)$, we must have $g \in T$ (note that T is really a set of natural numbers when this is formalized in RCA_0). A tree is infinite if it has infinitely many elements, and a tree is finitely-branching if, whenever f has domain $\{0, 1, ..., k\}$, $f \in T$, there are only finitely many $g \in T$ such that g has domain $\{0, 1, ..., k+1\}$ and $g(n) = f(n)$ for all $n \leq k$. A path in a tree is a sequence h such that, for all $k \in \mathbf{N}$, the restriction of h to the set of natural numbers less than k is an element of $\overline{T}$.

Theorem 2 (König's Lemma)**.** *The following is provable in* ACA_0*. If T is a finitely-branching infinite tree, there is a path in T.*

Proof. Let S be the set of finite sequences $f \in T$ such that there are infinitely many $g \in T$ that extend f (that is, their domains contain the domain of f and the restriction of g to the domain of f is equal to f). (When one writes the defining formula for S, using the encoding of finite sequences as numbers, the only quantifiers that appear are numerical and thus the arithmetical comprehension guaranteed by ACA_0 insures the existence of S.) Now for each finite sequence h defined for numbers less than n, define S_h to be the set consisting of numbers m such that the finite sequence g given by $g(j) = h(j)$ for $j < n$ and $g(n) = m$ is in S. (Again we are relying on our encoding of finite sequences to see that this set formation is really arithmetical.) Take f_0 to be the empty sequence, and for all n take f_{n+1} to be the finite sequence defined on natural numbers less than $n+1$ by $f_{n+1}(m) = f_n(m)$ for $m < n$ and $f_{n+1}(n) = 0$ if $S_{f_n} = \emptyset$; if $S_{f_n} \neq \emptyset$, take $f_{n+1}(n+1)$ to be the least element of S_{f_n}.

We can show by induction that $S_{f_n} \neq \emptyset$ for all n, as we know from our hypotheses that S_{f_0} is nonempty, and if $S_{f_n} \neq \emptyset$, we can conclude $f_{n+1} \in S$. Now we use the finite branching assumption to find the finite set F consisting of numbers j such that extending f_{n+1} by taking $n+1$ to j yields an element of T. By the definition of a tree, we know that, if g is any extension of f_{n+1}, either $g = f_{n+1}$ or $g(n+1) \in F$ (because the restriction of g to $\{0, ..., n+1\}$ must be in T). This implies that, if g is an extension of f_{n+1}, then either $g = f_{n+1}$ or g is an extension of some finite sequence h defined on $\{0, ..., n+1\}$ with h

[2]The analogy can be carried further: just as the algebraic closure of $\mathbf{Q}$ is not a finite extension, one cannot choose finitely many oracles from which every function from the natural numbers to itself can be computed. In fact one cannot even choose a countable number of oracles such that every function from the natural numbers to itself can be computed from one of the oracles, and more could be said in this vein. This is essentially because the number of instruction sets is countable, so there are only countably many functions that one can compute from a given oracle.

extending f and $h(n+1) \in F$. There are only finitely many such h's, so if each such h only has finitely many extensions in T, it follows that f_{n+1} only has finitely many extensions in T, which is impossible since $f_{n+1} \in S$. (Here we may appear to be using the set-theoretic notion of a finite union of finite sets being finite, but this is easily established within RCA_0, as if we have upper bounds $B_1, ..., B_s$ for all sets in the union, then $\max\{B_j\}$ is an upper bound for the union.) Consequently there is some h for which there are infinitely many extensions, and thus $h \in S_{f_{n+1}}$. This completes the induction.

It follows immediately from this that each f_n is in S, so each f_n is in T, and we can define a path f by declaring that $(j, n) \in f$ if and only if $(j, n) \in f_{j+1}$. †

It turns out that, over RCA_0, König's lemma is actually equivalent to ACA_0; that is, we can prove all of the axioms of ACA_0 if we assume RCA_0 and König's lemma. However, there is an important restriction of König's lemma that is not equivalent to ACA_0. A tree T is said to be binary if, whenever $f \in T$, the image of f is contained in $\{0, 1\}$.

Theorem 3 (Weak König's Lemma)**.** *The following is provable in* ACA_0*. Every infinite binary tree has a path.*

Of course the weak version follows immediately from the stronger. In fact it induces a strictly weaker subsystem of Z_2 than full König's lemma does; this is one of the Big Five subsystems and is called WKL_0. It is equivalent, for instance, to the statement that every continous function on $[0, 1] \subset \mathbf{R}$ is bounded, as well as to the Gödel completeness theorem. (As is generally true when we reason in reverse mathematics, the statements in the language of second-order arithmetic differ somewhat from the usual formalizations of these statements.) Note that WKL_0 is not equivalent to RCA_0; that is, weak König's lemma is not provable in RCA_0. See Simpson and Hirschfeldt for more complete discussions of variants of König's lemma.

5 Hindman's Theorem

Originally proved by Hindman in 1972, the theorem states

Theorem 4 (Hindman)**.** *If* $\{C_i\}_{i=1}^k$ *is some finite partition of* $\mathbf{N}$*, there is some* i *and an infinite set* $X \subset \mathbf{N}$ *such that, whenever* $x_1, ..., x_n$ *are distinct elements of* X*,* $\sum x_j \in C_i$*.*

A finite partition is called an instance of the Hindman problem, and X is a Hindman solution. Before proving Hindman's theorem, we will show a somewhat surprising result of Blass, Hirst, and Simpson. Before doing so, though, we need a technical lemma. To each natural number n we associate a set $\mathcal{B}(n) \subset \mathbf{N}$ such that $m \in \mathcal{B}(n)$ if and only if 2^m appears in the binary expansion of n. We say that a set X is separated if, whenever $n < m$ are in X, each element of $\mathcal{B}(n)$ is less than each element of $\mathcal{B}(m)$.

Lemma 5. *For any infinite set* $X \subset \mathbf{N}$*, there is an infinite separated set* Y *computable in* X *such that each element* y *of* Y *can be written as a distinct sum of (finitely many) elements* $z_1^y, ..., z_{n(y)}^y$ *of* X *such that the sets of* z^y*'s are disjoint. In particular,* $FS(Y) \subset FS(X)$*.*

Proof. We sketch the ideas and leave the details to the reader. Put $X_0 = X$. In what follows, we will refer to the $n-$th digit of a number in its binary expansion as its $n-$th coordinate. We will define a sequence of sets $\{X_n\}$ and a partial sequence of natural numbers $\{b_n\}$. For each $j > 0$, we divide into cases to define X_j :

1. If there is no element of X_{j-1} less than 2^{j+1} with k-th coordinate equal to 1 for some $k < j$, put $X_j = X_{j-1}$ and leave b_{j-1} undefined.

2. If $s < j$ is the smallest natural number such that there is some $u \in X_{j-1}$ with $u < 2^{j+1}$ and $s \in \mathcal{B}(u)$, and there are an odd number of elements of X_{j-1} with s-th coordinate equal to 1 and all lower coordinates equal to 0, say $a_1 < ... < a_n$, let X_j consist of elements of X_{j-1} with s-th coordinate equal to zero, elements of X_{j-1} with $s-$th coordinate equal to 1 and some lower coordinate equal to 1, and $a_{2k} + a_{2k+1}$ for $0 < k \leq n/2$. Put $b_{j-1} = a_1$.

3. If $s < j$ is the smallest natural number such that there is some $u \in X_{j-1}$ with $u < 2^{j+1}$ and $s \in \mathcal{B}(u)$, and there are an even number of elements of X_{j-1} with s-th coordinate equal to 1, say $a_1 < ... < a_n$, let X_j consist of elements of X_{j-1} with s-th coordinate equal to zero, elements of X_{j-1} with $s-$th coordinate equal to 1 and some lower coordinate equal to 1, and $a_{2k} + a_{2k+1}$ for $0 < k < n/2$. Put $b_{j-1} = a_1$.

4. If $s < j$ is the smallest natural number such that there is some $u \in X_{j-1}$ with $u < 2^{j+1}$ and $s \in \mathcal{B}(u)$, if there is an infinite number of elements of X_{j-1} with s-th coordinate equal to 1 and all lower coordinates equal to 0, say $a_1 < ...$, let X_j consist of elements of X_{j-1} with s-th coordinate equal to zero, elements of X_{j-1} with $s-$th coordinate equal to 1 and some lower coordinate equal to 1, and $a_{2k} + a_{2k+1}$ for $k > 0$. Put $b_{j-1} = a_1$.

The above list could be streamlined significantly, but the significant point is that we want to pair off elements so that, when we add them later, we reduce to sums of distinct elements of X. Losing a single element at the end of an even pairing is unimportant. The key is that it can be determined in finite time whether a b_m is defined for each b_m, but if this search ends and there is some element with coordinate 1 in the relevant position that was outside the bounds, that element is preserved until it enters the scope of a later search; it is not added to others.

It is worth noting that the above is actually, contrary to appearance, a computable (relative to X_{j-1}) construction of X_j: for any natural number m, we can compute the coordinates of each number in X_{j-1} less than 2^{j+1}; if none have a 1 coordinate in any position less than j, we know that $m \in X_j$ if and only if $m \in X_{j-1}$. Otherwise we know which k is under consideration and, if $m \in X_{j-1}$, $k \in \mathcal{B}(m)$, and k is not the least element of $\mathcal{B}(m)$, we know $m \in X_j$. We also know $m \in X_j$ if $m \in X_{j-1}$ and $k \notin \mathcal{B}(m)$. Otherwise we can enumerate the elements of X_{j-1} less than or equal to m and determine which of them have $k-$th coordinate equal to 1 and no lesser coordinates equal to 1, and m is in X_j if and only if it appears as one of the paired sums of these elements. This implies, by induction, that all X_n's are computable relative to X.

Note that the collection of b_n's is infinite, as otherwise we would have a finite bound on some $X_n = X_{n+1} = ...$ (as the scope function we chose, 2^{j+1}, becomes arbitrarily large), and it is clear from induction that each X_n is infinite. Also, at each coordinate k, note that eventually all elements of X_n will have 0 $k-$th coordinate; this can be seen by induction on k, as if each element of X_n has all lower coordinates equal to 0 whenever $n \geq N$, if all elements of x_N have $k-$th coordinate 0, then it is clear that all further X_n's only contain elements with $k-$th coordinate 0; otherwise, choose the least element $y \in x_N$ with $k-$th coordinate equal to 1; it is unaffected when b_m's are chosen corresponding to larger $k-$values, and thus once n is large enough that $y < 2^n$, it will be chosen as a b_m at some M; then it is clear that, when $n > M$, all elements of X_n have $k-$th coordinate equal to 0. Consequently if we let $c_n = b_n$ whenever $\lambda(b_n) > \mu(b_m)$ whenever $m < n$ and b_m is defined, c_n undefined otherwise, we obtain an infinite partial sequence of c_n's, for we have already established that the b_n's form an infinite collection and, given any N, all of the b_n's will eventually have first N coordinates equal to 0.

We leave it to the reader to verify that the collection of c_s's is computable from X (because all of the X_j's are computable from X; from each X_j we can verify whether there will be some b_j, and if there is a b_j, one may computably enumerate b_m's until either some m is found such that $n = b_m$ and $\lambda(n) > \mu(b_k)$ for $k < m$ or $n = b_m$ and there is some $k < m$ with $\lambda(n) \leq \mu(b_k)$ or neither of these occurs before some m is found such that b_m is defined and $\mu(m) > \lambda(n)$, in which case n is not a c_s; such an m must eventually arise because the c_s's are unbounded).

Now let Y be the collection of c_s's. We have just shown that Y is computable from X, and from its construction we know that every element of Y is a sum of distinct elements of X, with no element of X appearing in the sum for multiple elements of Y. Also by construction Y is separated and we are done. †

The lemma is due to Hindman.[3]

Theorem 6. *For any $S \subset \mathbf{N}$, there is a partition of $\mathbf{N}$ computable in S such that any Hindman solution X satisfies $S' \leq_T X$.*

Of course, S' denotes the Turing jump of S.

Proof. Choose a program $\mathcal{M}$ for a Turing machine with oracle S that computes a partial function $f : \mathbf{N} \to \mathbf{N}$ such that the range of f is the halting set for some fixed enumeration of the Turing machines with oracle S (the halting set is the set of natural numbers e for which φ_e^S halts on input e). Define $\lambda(n) = \min \mathcal{B}(n), \mu(n) = \max \mathcal{B}(n)$. A pair of natural numbers (r, s) is said to be a gap in n if $r, s \in \mathcal{B}(n)$ and $r < t < s \implies t \notin \mathcal{B}(n)$. Such a gap is short if there is some $x \leq r$ such that $x \in S'$ but $\mathcal{M}$ does not halt on input x in s or fewer steps. And it is very short if there is some $x \leq r$ such that $x \in S'$, $\mathcal{M}$ halts on input x in $\mu(n)$ or fewer steps, but $\mathcal{M}$ does not halt in s or fewer steps. Of course every very short gap is a short gap. Let $SG(n)$ denote the number of short gaps in n, and let $VSG(n)$ be the number of very short gaps in n.

We claim that VSG is a computable function relative to X. It is clear that $\mathcal{B}(n)$ is computable. To determine whether a gap (r, s) is very short, on each $x \leq r$, run $\mathcal{M}$ for at most s steps (of course we stop before then if $\mathcal{M}$ halts in fewer steps). If $\mathcal{M}$ has not halted, continue running $\mathcal{M}$ so that it has performed, in total, no more than $\mu(n)$ steps; if a 1 is then output, we know that (r, s) is a very short gap. There are only finitely many $x \leq r$, so we can repeat this for all such x. Performing this for all gaps in n, we determine exactly which gaps are very short and we can then count them. This shows the computability of VSG.

Now we color $\mathbf{N}$ by declaring the set of n such that $VSG(n)$ is even to be one color and the set of n such that $VSG(n)$ is odd another color. We know that this coloring is computable relative to X. Let X be a solution for this Hindman problem. Then

[3]It should be said that the original proof in [7] is much shorter and of a different character; I wrote this one before coming across Hindman's original argument.

we can use the lemma to compute a solution Y relative to X with disjoint, increasing binary expansion sets. It will suffice to show that we can compute $S' \leq_T Y$.

We claim that, for all m that can be written as a finite sum of elements of Y, $SG(m)$ is even. We can choose some $n \in Y$ such that $n > m$ and, for all $x \leq \mu(m)$, if $x \in S'$, then $\mathcal{M}$ halts on x in $\lambda(n)$ or fewer steps. This is simply because the restriction of λ to Y is unbounded by choice of Y. The gaps of $n+m$ consist of the gaps of n, the gaps of m, and the gap $(\mu(m), \lambda(n))$ (as m is a sum of elements of Y less than n, all of which have disjoint binary expansions, we know $\mu(m) < \lambda(n)$). A gap in n is short if and only if the same gap is short in $n+m$, and the same holds for m. Further, any short gap in m is very short in $n+m$ by choice of n. And a gap in n is very short in n if and only if it is very short in $n+m$, as $\mu(n+m) = \mu(n)$. Finally, wee see that the gap $(\mu(m), \lambda(n))$ is not very short in $m+n$ directly from the choice of n. Consequently $VSG(n+m) = SG(m) + VSG(n)$. But $VSG(n+m), VSG(n)$ have the same parity because they are both finite sums of elements in Y and hence have the same color by the fact that Y is a solution to the Hindman problem. This proves that $SG(m)$ is even.

Now, if $n, m \in Y$ with $m < n$, $x \leq \mu(m)$, we claim that if $x \in S'$, our machine $\mathcal{M}$ halts on x in $\lambda(n)$ or fewer steps. The claim is the statement that $(\mu(m), \lambda(n))$ is not short. If it were, we would have $SG(n+m) = SG(n) + SG(m) + 1$, impossible since the left side is even and the right side is odd. So we can compute S' by finding $n, m \in Y$ with $x \leq \mu(m) < \lambda(n)$ (we can do this relative to X) and running $\mathcal{M}$ on input x for $\lambda(n)$ steps. If a 1 is output, $x \in S'$; otherwise, $x \notin S'$. †

Our prior discussion of $\omega-$models of ACA_0 then gives us the following result:

Corollary 7. *Every $\omega-$model of Hindman's theorem is a model of* ACA_0.

And when we abstract from Turing jumps to their analogue defined using universal lightface Π^0_1 formulae:

Corollary 8. *All axioms of* ACA_0 *are provable in* HT.

This is the best lower bound of the strength of the subsystem of Z_2 induced by Hindman's theorem currently known. But we now turn to proving Hindman's theorem itself in a suitable subsystem of Z_2; in doing so, we will obtain computability bounds for the necessary operations, and this will give us sufficient criteria for a collection of subsets of $\mathbf{N}$ to form an $\omega-$model of HT. The proof we give is Hindman's original argument in [6]; the computability- and proof-theoretic analysis was first established by Blass, Hirst, and Simpson in [2], and we closely follow their argument.

We must now introduce some notation. For each $X \subset \mathbf{N}$ we write $FS(X)$ for the set of natural numbers obtainable as finite (nonempty) sums of distinct elements of X. We will say that a sequence $\langle x_i \rangle_1^\infty$ is separated if it is strictly increasing and its image is separated. In this case we define a map $\tau : FS(\langle x_i \rangle_1^\infty) \to \mathbf{N}$ by $\tau(\sum_{j \in F} x_j) = \sum_{j \in F} 2^{j-1}$. (The -1 in the exponent is just to correct for the fact that we chose our sequences to begin with index 1.) It is easy to see that τ is bijective and preserves addition. All of the sequences we address in the following argument are strictly increasing, so we will not state this explicitly each time. Note that an increasing sequence has the same Turing degree as its image, for if we are given the sequence as an oracle, we can compute membership of a given number m in the image by simply enumerating the values of the sequence until an output is obtained that is greater than or equal to m. Conversely we can compute the n-th term in the sequence from the set from dovetailing computations until n 1's are output, say the greatest is u; then the computation continuous on natural numbers less than u until they have all converged, and the n-th number to output 1 is then selected. (Note here that we are implicitly using a pairing function when discussing the Turing degree of the sequence.)

If X is a separated infinite set, using the same argument as in the previous lemma but replacing the coordinates with respect to binary powers with coordinates with respect to an increasing enumeration $x_1, x_2, ...$ of X, we obtain the following result:

Lemma 9. *If $\langle x_i \rangle_1^\infty$ is separated, $\langle y_i \rangle_1^\infty$ a sequence such that $FS(\langle y_i \rangle) \subset FS(\langle x_i \rangle)$, there is an increasing sequence $\langle z_i \rangle_1^\infty$ computable in the join of $\langle x_i \rangle, \langle y_i \rangle$ such that $\tau(\langle z_i \rangle)$ is separated and $FS(\langle z_i \rangle) \subset FS(\langle y_i \rangle)$.*

Note that we need the oracle $\langle x_i \rangle$ to compute coordinates with respect to $\langle x_i \rangle$; previously, $\mathcal{B}(n)$ was simply a computable function. If we restate this in terms of subsystems of second-order arithmetic, using the extended notion of computability mentioned earlier, we find that it is provable in RCA_0.

Lemma 10. *The following is provable in* RCA_0. *If $\langle x_i \rangle_1^\infty$ is separated, $\langle y_i \rangle_1^\infty$ a sequence such that $FS(\langle y_i \rangle) \subset FS(\langle x_i \rangle)$, there is an increasing sequence $\langle z_i \rangle_1^\infty$ recursive in the join of $\langle x_i \rangle_1^\infty$ and $\langle y_i \rangle$ such that $\tau(\langle z_i \rangle)$ is separated and $FS(\langle z_i \rangle) \subset FS(\langle y_i \rangle)$.*

We are here using the correspondence between 'Y is computable in X' and 'given X, the existence of Y is provable in RCA_0'. But if one examines our proof of the original lemma, it is clear that it can be formalized entirely as a construction of Y in terms of X within RCA_0; we just stated it in terms of a computability relation. (This is why we went to the trouble of bounding our quantifiers when recursively defining the X_j's.) So this is perhaps closer to what we proved directly than the original statement we gave. Now we move into the main part of the argument. We say that a subset of the 'real' natural numbers ω is arithmetical if it is computable from $\emptyset^{(n)}$ for some natural number n.

Theorem 11. *The following is provable in* ACA_0. *Let* $k > 0$ *and suppose we have an indexed collection of sets* $A(i,n)$ *for* $0 < i \leq k$ *and* $n > 0$ *such that* $A(i,n+1) \subset A(i,n)$ *for all* i, n. *Then there is some* $S \subset \{1,...,k\}$, *a sequence* $\langle x_j \rangle_1^\infty$ *computable in* $\langle A(i,n) \rangle$, *and some* M *such that, whenever* $n \geq M$, $\langle y_m \rangle_1^\infty$ *a sequence in* $\langle A(i,n) \rangle$ *with* $FS(\langle y_m \rangle) \subset FS(\langle x_m \rangle)$, *we have* $FS(\langle y_m \rangle) \cap A(i,n) \neq \emptyset$ *if and only if* $i \in S$.

Proof. The proof is by induction, though we must be somewhat careful about what the induction hypothesis is. Fix k and consider $1 \leq \mu \leq k$. We want to show by induction that there is some sequence $\langle x_{j\mu} \rangle_j$ with some M_μ such that, whenever $n, r \geq M$, $1 \leq i \leq \mu$, $\langle y_m \rangle_1^\infty$ a sequence with $FS(\langle y_m \rangle) \subset FS(\langle x_m \rangle)$, we have $FS(\langle y_m \rangle) \cap A(i,n) \neq \emptyset$ if and only if $FS(\langle y_m \rangle) \cap A(i,r) \neq \emptyset$.

If $\mu = 1$ and there is some number n and a sequence $\langle x_j \rangle_1^\infty$ such that $FS(\langle x_n \rangle) \cap A(1,n) = \emptyset$, put $M_1 = n$. Otherwise put $M_1 = 1$ with $\langle x_j \rangle_1^\infty$ some arbitrary increasing sequence computable in $\langle A(1,n) \rangle$. It is not too difficult to see that these M_1 must satisfy the conclusions, for if $FS(\langle x_n \rangle) \cap A(1,n) = \emptyset$, it is clear that the intersection of any subset of $FS(\langle x_n \rangle)$ with $A(1,n)$ is the empty set, and the fact that the sets are decreasing implies that all further $A(1,m)$ are also empty. In the case where there is no such $\langle x_j \rangle_1^\infty$, note that every possible $\langle y_m \rangle$ is computable from $\langle A(i,n) \rangle$, so from the assumptions for this case we have $FS(\langle y_m \rangle) \cap A(1,n) \neq \emptyset$.

Now for the induction. Suppose the claim is valid for μ. To see that it is valid for $\mu + 1$ whenever $\mu < k$, note that we have already obtained some $\langle x_{n\mu} \rangle, M_\mu$ that apply when $1 \leq i \leq \mu$. If there is some sequence $\langle y_m \rangle$ recursive in $\langle A(i,n) \rangle_{i,n}$ with $FS(\langle y_m \rangle) \subset FS(\langle x_n \rangle)$ and $FS(\langle y_m \rangle) \cap A(\mu+1, N) = \emptyset$ for some N, choose $\langle x_{n,\mu+1} \rangle = \langle y_m \rangle$ and put $M_{\mu+1} = M_\mu + N$. If there is no such $\langle y_m \rangle$, let $\langle x_{n,\mu+1} \rangle = \langle x_{n\mu} \rangle$ and $M_{\mu+1} = M_\mu$. One may check that these have the requisite properties by the same reasoning as in the base case. This proves that the result holds when $\mu = k$, as desired. Then define S by

$$S = \{s \in \mathbf{N} : s \leq k \wedge (\exists c \in \mathbf{N})(\forall d \in \mathbf{N})[d \geq c \implies FS(\langle y_m \rangle) \cap A(s,n) \neq \emptyset]\}.$$

The formula is arithmetical and comprehension applies, so we are done.

†

Note that, if we dropped all requirements involving computability from the theorem, the result would not be provable in ACA_0 by the same method, for there would truly be no way of reducing the set quantifiers to number quantifiers. The result would, however, be provable in ACA.

Now whenever we have a finite partition α of $\mathbf{N}$ consisting of sets $A_1, ..., A_m$, we associate double sequences of sets $F_\alpha, F'_\alpha, U_\alpha$ by declaring, for $0 < k < n$, that $F'_\alpha(k,n)$ is the set of natural numbers x such that $x \geq n$ and there is some i such that $\{k, x, x+k\} \subset A_i$. We similarly declare $F_\alpha(1,n) = F'_\alpha(1,n)$ for all $n > 1$ and subsequently $F_\alpha(k,n) = F'_\alpha(k,n) \setminus \cup_{j=1}^{k-1} F'_\alpha(j,n)$. Then, for $i = 1, ..., m$, we declare $U_\alpha(i,n) = (A_i \cap \{x \in \mathbf{N} : x \geq n\}) \setminus \cup_{k=1}^{n-1} F_\alpha(k,n)$.

Note that we may equivalently regard the computability-theoretic strength of a partition α consisting of $A_1, ..., A_m$ as the join of their Turing degrees or the characteristic function of the set of pairs (i,n) where $n \in A_i$.

Theorem 12. *The following is provable in* ACA_0. *If* α *is a partition consisting of* $A_1, ..., A_q$, *such that, for every* $n \in \mathbf{N}$ *and every* $\langle t_\sigma \rangle_1^\infty$ *recursive in* α, $\cup_{k=1}^{n-1} F_\alpha(k,n)$ *does not contain* $FS(\langle t_\sigma \rangle)$, *then there are sequences* $\langle x_n \rangle$, $\langle \mu_n \rangle$ *(both sequences of numbers), sequences* $\langle y_{n,m} \rangle_m$ *recursive in* α *for each* n, *and a sequence of sets* $\langle U_{n,m} \rangle_m$ *recursive in* α *for each* n *such that the following hold:*

1. *For each* n, $\langle y_{n,m} \rangle_m$ *is separated.*
2. *If* $p \geq \mu_n$, $n \in \mathbf{N}$, *and* $\langle z_m \rangle$ *is a sequence recursive in* α *such that* $FS(\langle z_m \rangle) \subset FS(\langle y_{n,m} \rangle)$, *then* $FS(\langle z_m \rangle) \cap U_{n,p} \neq \emptyset$.
3. *For each* n, *there is some* i *such that, if* $p \geq \mu_n$, *then* $U_{n,p+1} \subset U_{n,p} \subset A_i$.
4. *If* $n > 0$, *then* $\mu_n > \sum_1^n x_j$.
5. *If* $n > 0$ *and* $p \geq \mu_n$, *then* $U_{n,p} \subset U_{n-1,p}$.
6. *If* $n > 0$, $p \geq \mu_n$, $x \in U_{n,p}$, *then* $x + x_n \in U_{n-1,\mu_{n-1}}$.

Proof. It is clear that the $U_\alpha(i,n)$ defined above satisfy the hypotheses of Theorem 11, so we obtain appropriate M, S, and $\langle w_m \rangle$. Suppose for contradiction that $S = \emptyset$. Then $FS(\langle w_m \rangle_1^\infty) \cap U_\alpha(i,M) = \emptyset$ for all i (taking $\langle y_k \rangle$ in Theorem 11 to be $\langle w_m \rangle$ itself). From the definition of the U_α's, we see that the union $\cup_i U_\alpha(i,M)$ contains all natural numbers that are greater than or equal to M and not contained in $\cup_1^{M-1} F_\alpha(i,M)$. If we define the set $\langle c_k \rangle_1^\infty$ to be the elements of $\langle w_m \rangle$ greater than M, we then find that each element of $FS(\langle c_k \rangle)$ is greater than M and none are in $\cup_i U_\alpha(i,M)$, whence we find $FS(\langle c_k \rangle) \subset \cup_1^{M-1} F_\alpha(i,M)$, which

is impossible by assumption. Thus $S \neq \emptyset$.

Fix some $i \in S$ (say i is the least element of S) and put $x_0 = 0, \mu_0 = M$, $\langle y_{0,m} \rangle_1^\infty$ a separated sequence computable in $\langle U_\alpha(j,n) \rangle$ (and thus computable in α, as one may easily see that $\langle U_\alpha(j,n) \rangle$ is computable in α) with $FS(\langle x_{0,m} \rangle) \subset FS(\langle w_m \rangle)$. (This is possible by Lemma 5, which is provable in RCA_0.) Whenever $p \geq \mu_0$ put $U(0,p) = U_\alpha(i,p)$ (one may choose $U(0,q)$ for $q < \mu_0$ arbitrarily; for concreteness we say $U(0,q) = \emptyset$ in such cases). It is easy to check that these designations satisfy all of the requirements 1-6.

We will now recursively define our sequences. Suppose we have defined our sequences $\langle x_n \rangle$, $\langle \mu_n \rangle$, $\langle y_{n,m} \rangle_m$, $\langle U_{n,m} \rangle_m$ for $k < n$. We now want to define them at n. Because $\langle y_{n-1,m} \rangle_m$ is separated by assumption, it has some natural map $\tau : FS(\langle y_{n-1,m} \rangle_m) \to \mathbf{N}$. Suppose $p \geq \mu_{n-1}$. We now claim that, for each sequence $\langle b_m \rangle$ recursive in α, $FS(\langle b_m \rangle_m) \cap \tau(U(n-1,p)) \neq \emptyset$. Suppose for contradiction that there is such a $\langle b_m \rangle$ such that the intersection is empty. By Lemma 5 (which, as we mentioned before, is provable in RCA_0 and thus certainly provable in ACA_0), we may assume without loss of generality that $\langle b_m \rangle$ is separated. As τ is a bijection, it has an inverse (which may be easily defined within ACA_0), so taking the image of both sides over τ^{-1} gives $\tau^{-1}(FS(\langle b_m \rangle_m)) \cap U(n-1,p) = \emptyset$. Because $\langle b_m \rangle$ is separated, we can rewrite the left side as $FS(\langle \tau^{-1}(b_m) \rangle) \cap U(n-1,p)$, so

$$FS(\langle \tau^{-1}(b_m) \rangle) \cap U(n-1,p) = \emptyset.$$

But of course each $\tau^{-1}(b_m)$ is contained in $FS(\langle y_{n-1,m} \rangle)$, and thus $FS(\langle \tau^{-1}(b_m) \rangle) \subset FS(\langle y_{n-1,m} \rangle$ because $\langle b_m \rangle$ is separated. But this contradicts our assumption that condition 2 holds at $n-1$. This proves that $FS(\langle b_m \rangle_m) \cap \tau(U(n-1,p)) \neq \emptyset$.

Suppose now for contradiction that there are arbitrarily long intervals $\{t \in \mathbf{N} : a \leq t \leq b\}$ that have nonempty intersection with $\tau(U(n-1,\mu_{n-1}))$. We can then construct a sequence $\langle a_k \rangle_1^\infty$ recursively in $U(n-1,\mu_{n-1})$ (and thus recursively in α) by taking a_1 to be the least element of $\mathbf{N} \backslash U_{n-1,p}$ and, whenever $a_1, ..., a_{s-1}$ have been defined, taking a_s to be the least natural number greater than all $a_1, ..., a_{s-1}$ such that the interval $\{t : a_s \leq t \leq a_s + \sum_{u=1}^{s-1} a_u\}$ is disjoint from $\tau(U(n-1,\mu_{n-1}))$. (Such an a_s exists precisely by our assumption that there are arbitrarily long intervals disjoint from $\tau(U(n-1,\mu_{n-1}))$.) It is then obvious from the construction that $FS(\langle a_s \rangle) \cap \tau(U_{n-1,p}) = \emptyset$, which, as we have shown above, is impossible. So we have some least upper bound B_0 on the length (difference between endpoints) of intervals disjoint from $\tau(U(n-1,\mu_{n-1}))$; let $B = B_0 + 3$ (this choice of B is computable in $U_{n-1,p}$ because one may computably find the least number with a property that can be computably verified). Now we know that, for all $x \in \mathbf{N}$, $\{x+1, ..., x+B\} \cap \tau(U_{n-1,\mu_{n-1}}) \neq \emptyset$ (as the left side is an interval of length greater than B_0).

Put $\mu'_n = \max\{\mu_{n-1}, \tau^{-1}(B) + 1 + \sum_{j=1}^{n-1} x_j\}$, and take r to be the largest integer such that $r - 1 \in \mathcal{B}(B)$ (i.e. $2^{r-1} \leq B$). Note that these must be recursive in the join of $\langle y_{n-1,m} \rangle_m$ and $U_{n-1,\mu_{n-1}}$. For each $1 \leq j \leq B$ and $p \geq \mu'_n$, put

$$V(j,p) = \{x \in \tau(U_{n-1,p}) : 2^r | x \wedge x + j \in \tau(U_{n-1,\mu_{n-1}})\}.$$

Define also $V(0,p) = \{x \in \tau(U_{n-1,p}) : 2^r \nmid x\}$. From our prior discussion of the properties of B, we have that, for each p, $\tau(U_{n-1,p}) = \cup_{j=0}^B V(j,p)$ (since, if $x \in \tau(U_{n-1,p})$ and $2^r | x$, $\{x+1, ..., x+B\}$ is not disjoint from $\tau(U_{n-1,\mu_{n-1}})$ and so there is some $1 \leq j \leq B$ such that $x \in V(j,p)$. Because we assumed that $U_{n-1,p+1} \subset U_{n-1,p}$, we have $V(j,p+1) \subset V(j,p)$. If $p < \mu'_n$, let $V(j,p) = V(j,\mu'_n)$. We can now apply Lemma 11 to each $\langle V(j,p) \rangle_p$. This gives us a set $S' \subset \{0, ..., B\}$, a sequence $\langle h_k \rangle_1^\infty$, and a natural number T (all with the complexity properties stated in Lemma 11) such that, whenever $p \geq T$ and $\langle c_m \rangle$ any sequence recursive in α such that, if $FS(\langle c_m \rangle) \subset FS(\langle h_k \rangle)$, then $FS(\langle c_m \rangle) \cap V(j,p) \neq \emptyset$ if and only if $j \in S'$. Again using Lemma 10, we may assume that $\langle h_k \rangle$ is separated.

We claim that $S' \neq \emptyset$. For if $S' = \emptyset$, we have that whenever $p \geq T$, $\langle c_m \rangle$ recursive in α such that $FS(\langle c_m \rangle) \subset FS(\langle h_r \rangle)$, then $FS(\langle c_m \rangle) \cap V(j,p) = \emptyset$ for all j, so

$$FS(\langle c_m \rangle) \cap \tau(U_{n-1,p}) = FS(\langle c_m \rangle) \cap (\cup_j V(j,p)) = \emptyset,$$

because we know $\tau(U_{n-1,p}) = \cup_{j=0}^B V(j,p)$. But we have previously shown this is impossible. (It is easy to see that this reasoning goes through in ACA_0.) Hence $S' \neq \emptyset$, as desired. Moreover $0 \notin S'$, for the set that is obtained by removing the least $r+1$ elements of $\langle h_r \rangle$ is recursive in α (since it is recursive in $\langle h_r \rangle$, so recursive in the double sequence $\langle V(j,q) \rangle$, which is itself recursive in $U_{n-1,p}$, which is recursive in α by its construction) and, by the separation of $\langle h_r \rangle$, all of its elements are divisible by 2^r, and thus all sums of its elements are divisible by 2^r, so its finite sum set is disjoint from $V(0,p)$ and the claim follows from choice of S'. Consequently we have that S' contains some positive integer.

Let w be the least element of S'. Define $x_n = \tau^{-1}(w)$ and let $\mu_n = \max\{\mu_{n-1}, T\}$. For all m define $y_{n,m} = \tau^{-1}(h_m)$ and, for $p \geq \mu_n$, define $U(n,p) = \tau^{-1}(V(w,p))$.

Verifying that these satisfy the requisite properties is now a straightforward exercise (as, of course, the construction was made so that this part of the argument is nearly trivial). We may assume $n > 0$ as we have already dealt with the base case. By the definition of τ and the fact that both $\langle r_k \rangle$ and $\langle y_{n-1,m} \rangle_m$ are separated, condition 1 holds at n. Condition 2 just follows from the fact that

$$\begin{aligned}\tau(FS(\langle z_m \rangle) \cap U_{n,p}) &= \tau(FS(\langle z_m \rangle)) \cap \tau(U_{n,p}) \\ &= FS(\langle \tau(z_m) \rangle) \cap V(w,p) \neq \emptyset\end{aligned}$$

by choice of w. As for condition 3, because $V(w,p+1) \subset V(w,p)$ whenever $p \geq \mu_n$, we have that $U_{n,p+1} \subset U_{n,p}$ for such p. We obtained an i for the base case, and as we claim that that i works for all n, we may assume that condition 3 holds at $n-1$ with that i. Then, if $p \geq \mu_n$, we get that $\tau(U_{n,p}) = V(w,p) \subset \tau(U_{n-1,p})$, and therefore $U_{n,p} \subset U_{n-1,p} \subset A_i$, where the last inclusion uses the fact that $\mu_{n-1} \leq \mu_n$. So much for condition 3. Now condition 4 follows from the definition of μ'_n, since τ and τ^{-1} are increasing functions and $w \leq B$. For condition 5, we again use the fact that $U_{n,p}$ is contained in the domain of τ and $\tau(U_{n,p}) = V(w,p) \subset \tau(U_{n-1,p})$ from the construction of the V's whenever $p \geq \mu_n$. Regarding condition 6, note that whenever $p \geq \mu_n$ and $x \in U_{n,p}$, our choice of r and the fact that $w \leq B$ forces $\tau(x) \in V(w,p)$ to imply $\mathcal{B}(\tau(x)) \cap \mathcal{B}(\tau(x_n)) = \emptyset$, as $\mathcal{B}(\tau(x))$ contains nothing less than r by definition of $V(w,p)$. This means that, when written as finite sums of distinct elements of $\langle y_{n-1,m} \rangle_m$, the expressions for x and x_n involve disjoint subsets of $\langle y_{n-1,m} \rangle_m$ (this comes directly from the definition of τ). But this gives us that $x + x_n \in FS(\langle y_{n-1,m} \rangle_m)$ and, in particular, $\tau(x + x_n) = \tau(x) + \tau(x_n)$. So we have a sum of w and some element of $V(w,p)$, which is in $U_{n-1,p}$ by construction of V. The proof is complete.

†

Lemma 13. *The following is provable in* ACA_0. *If $A_1, ..., A_q$ form a partition α of $\mathbf{N}$ such that, for every $n \in \mathbf{N}$ and every sequence $\langle y_m \rangle_1^\infty$ recursive in α, $FS(\langle y_m \rangle)$ is not contained in $\cup_{k=1}^{n-1} F_\alpha(k,n)$, there is some $1 \leq 1 \leq q$ and a sequence $\langle x_n \rangle_1^\infty$ in $\mathbf{N}$ such that $FS(\langle x_n \rangle) \cap A_i = \emptyset$.*

Proof. We take i and $\langle x_n \rangle_1^\infty$ from Theorem 12. Let F be a finite subset of $\mathbf{N}$. We wish to show that $\sum_{j \in F} x_j \notin A_i$. Take t to be the smallest element of F and r to be the largest element. Pick some $x \in U_{r,\mu_r}$, where μ_r is also as in Theorem 12 (note that $U_{r,\mu_r} \neq \emptyset$ is an immediate consequence of condition 3 in the theorem). Let $F_1 = \{r\}$ and, for $m > 1$, let F_m be the union of F_{m-1} and the singleton set containing the greatest element of $F \backslash F_m$ if such an element exists; else, let $F_m = F_{m-1}$. We will show by induction that, for all m, $x + \sum_{n \in F_m} x_n \in U_{t-1,\mu_{t-1}}$. It is clear from Theorem 12 that $x + r \in U_{t-1,\mu_{t-1}} \subset U_{r-1,\mu_r}$, so the base case is done. And if $F_m = F_{m-1}$, where the statement holds for $m-1$, it obviously holds for m as well, so there is nothing to prove here. So we may assume that $F_m = F_{m-1} \cup \{s\}$, where $s \notin F_{m-1}$. If t' is the least element of F_{m-1}, we know that $x + \sum_{n \in F_{m-1}} \in U_{t'-1,\mu_{t'-1}}$ by the inductive hypothesis. By Theorem 12, we have $U_{t'-1,\mu_{t'-1}} \subset U_{s,\mu_s}$, so $x + \sum_{n \in F_m} x_n = x_s + (x + \sum_{n \in F_{m-1}} x_n) \in U_{s-1,\mu_{s-1}}$, also by the theorem. This completes the induction. Note that $F_m = F$ for sufficiently large m, as we would otherwise have an infinitely long strictly decreasing sequence, which is impossible. So this proves that $x + \sum_{n \in F} x_n \in U_{t-1,\mu_{t-1}} \subset A_i$, and the theorem also gives us that $x \in A_i$.

Note that $x \in U_{r,\mu_r} \subset U_{0,\mu_r} = U_\alpha(i,\mu_r)$ from the definition of U_{0,μ_r}. And, because $\cup_{k=1}^{n-1} F_\alpha(k,n) = \cup_{k=1}^{n-1} F'_\alpha(k,n)$ for all n, $U_\alpha(i,\mu_r)$ is disjoint from $\cup_{k=1}^{\mu_r - 1} F'_\alpha(k,\mu_r)$. Hence $U_\alpha(i,\mu_r)$ is disjoint from $F'_\alpha(\sum_{n \in F} x_n, \mu_r)$ (note that $\sum_{n \in F} x_n < \mu_r$ by construction of μ_r). Hence $x \notin F'_\alpha(\sum_{n \in F} x_n, \mu_r)$. And this immediately implies that $\{x, \sum_{n \in F} x_n, x + \sum_{n \in F} x_n\}$ is not contained in A_i. By our prior remarks, this is only possible if $\sum_{n \in F} x_n \notin A_i$, as desired. †

Let us pause to remark here that we have done very little to modify the substance of Hindman's proof to this point: the proof was already essentially arithmetical. The only modifications we have made are restrictions to recursive sequences, which do little to alter the character of the arguments.

Lemma 14. *The following is provable in* ACA_0. *If α is a partition of $\mathbf{N}$ consisting of $A_1, ..., A_q$, there is some $n \in \mathbf{N}$ and some sequence $\langle x_m \rangle_1^\infty$ such that $FS(\langle x_m \rangle) \subset \cup_{k=1}^{n-1} F_\alpha(k,n)$.*

Proof. We induct on q. If $q = 1$, there is little to prove, as $F_\alpha(1,2)$ consists of all natural numbers greater than 1. Now assume for contradiction that $q > 0$, the conclusion holds for all partitions consisting of $q-1$ sets, and the conclusion does not hold for α. By Lemma 13 and 10, we obtain some $1 \leq i \leq q$ and a separated sequence $\langle y_m \rangle_1^\infty$ such that $FS(\langle y_m \rangle) \cap A_i = \emptyset$. Write $\tau : FS(\langle y_m \rangle) \to \mathbf{N}$ for the natural map and assume without loss of generality that $i > 1$. For all $1 < j \leq q$, put $B_j = \tau(A_q)$, and put $B_1 = \{0\} \cup \tau(A_1)$. It is clear that $B_i = \tau(A_i) = \emptyset$ and the B_j's partition $\mathbf{N}$, so they form a partition β of $\mathbf{N}$ consisting of $q-1$ sets (some of which may be empty). By the inductive hypothesis, we can find a sequence $\langle z_m \rangle_1^\infty$ and a number r such that $FS(\langle z_m \rangle) \subset \cup_{k=1}^{r-1} F_\beta(k,r)$; as usual, we can assume without loss of generality that it is separated, and by simply

excluding the first $r+1$ elements and re-indexing we may assume without loss of generality that 2^r divides all of the z_m. Put $n = \tau^{-1}(r), x_m = \tau^{-1}(z_m)$ for all positive integers m. Naturally, we claim that $FS(\langle x_m \rangle) \subset \cup_{k=1}^{n-1} F_\alpha(k,n)$. As τ takes the sum of distinct elements of $\langle y_m \rangle$ to a sum of corresponding powers of 2 and $\langle z_m \rangle$ is separated, we know that, when we apply τ to any element c of $FS(\langle x_m \rangle)$, we obtain the corresponding element of $FS(\langle z_m \rangle)$. Then we clearly have $\tau(c) \in FS(\langle z_m \rangle) \subset \cup_{k=1}^{r-1} F_\beta(k,r)$, whence we have $\{k, \tau(c), \tau(c)+k\} \subset B_j$ for some $0 < k < r$ and $1 \leq j \leq q$. Because $k, \tau(c) > 0$, we can conclude that $\{k, \tau(c), \tau(c)+k\} \subset \tau(A_j)$ for such a j. for So $\{\tau^{-1}(k), c, c+\tau^{-1}(k)\} \subset A_j$, as our assumption that $2^r | z_m$ for all m implies $2^r | \tau(c)$, and each element of $\mathcal{B}(k)$ is less than or equal to 2^r since $k < r$, so $\mathcal{B}(\tau(c)) \cap \mathcal{B}(k) = \emptyset$ and thus τ^{-1} preserves the addition. Moreover $c \geq n$ because τ, τ^{-1} are order-preserving and $\tau(c) \geq \tau(n) = r$ from the definition of F'_β. This proves that $c \in F'_\alpha(k,n) \subset \cup_{\kappa=1}^{n-1} F'_\alpha(\kappa, n) = \cup_{\kappa=1}^{n-1} F_\alpha(\kappa, n)$. As c was arbitrary, this proves $FS(\langle x_m \rangle) \subset \cup_{\kappa=1}^{n-1} F_\alpha(\kappa, n)$, a contradiction.[4] †

Now we turn to the more interesting analysis of where ACA_0 does not fully carry us through Hindman's proof. For each $X \subset \mathbf{N}$ and $j \in \mathbf{N}$, write $X_j = \{n : (n,j) \in X\}$ and $X^j = \{n : \exists m \exists i (n = (m,i) \wedge n \in X \wedge i < j)\}$, where $(\cdot,\cdot)$ denotes an aforementioned pairing map. The system ACA_0^+ consists of ACA_0 and all formulae of the form $\exists X \forall j \forall n (n \in X_j \iff \varphi(n, X^j))$, where φ is an arithmetical formula not containing X as a free variable. This axiom extends arithmetical comprehension by allowing us to essentially form a set that contains information about satisfying an infinite family of arithmetical formulae simultaneously. We will see that this is exactly what we need to carry out the rest of the argument. In the context of $\omega-$models, this is equivalent to stating that, if X is contained in the model, the $\omega-$th jump of X is also in the model. (We do not define the ω-th jump, as it is not particularly relevant to the proof-theoretic analysis, but the aforementioned references on computability address it.)

Lemma 15. *The following is provable in* ACA_0^+. *If $A_1, ..., A_q$ form a partition α of* $\mathbf{N}$, *there is a function $f_\alpha : \mathbf{N} \to \mathbf{N}$ such that, for each $r > 0$, there is some $1 \leq i \leq q$ and $\langle y_j \rangle_{j=1}^r$ such that the following hold:*

1. *$FS(\langle y_j \rangle) \subset A_i$.*
2. *$\langle y_j \rangle$ is separated.*
3. *For all $1 \leq j \leq r$, $y_j \leq f_\alpha(j)$.*

Proof. We will inductively define a sequence of partitions of $\mathbf{N}$. Let $\alpha_1 = \alpha$. Suppose $n \geq 1$ and α_n has been defined. By Lemma 14, we can find some $p_n \in \mathbf{N}$ and a sequence $\langle x_{n,m} \rangle_{m=1}^\infty$ such that $FS(\langle x_{n,m} \rangle_m) \subset \cup_{k=1}^{p_n - 1} F_{\alpha_n}(k, p_n)$. Write $\tau_n : FS(\langle x_{n,m} \rangle_m) \to \mathbf{N}$ for the natural map. We can assume without loss of generality that $\langle x_{n,m} \rangle_m$ is separated, and by removing some initial terms if necessary, we may assume that, for all m, each element of $\mathcal{B}(p)$ is less than each element of $\mathcal{B}(\tau(x_{n,m}))$. For $1 < k < p_n$, let $C_k = \tau_n(F_{\alpha_n}(k, p_n))$, and put $C_1 = \{0\} \cup \tau_n(F_{\alpha_n}(1, p_n))$ (recall that images of sets over functions are defined in ACA_0 via the formula $\exists m (f(m) = n)$). Note that $F_{\alpha_n}(k, p_n)$ may not be contained in the domain of τ_n. But by the containment $FS(\langle x_{n,m} \rangle_m) \subset \cup_{k=1}^{p_n - 1} F_{\alpha_n}(k, p_n)$, we have that the C_k's form a partition of $\mathbf{N}$; call this partition α_{n+1}. Now, for each m, define $T(n,m) = \tau_n^{-1}(m)$; this requires ACA_0^+, as for any fixed n the sequence $T(n,\cdot)$ is definable in ACA_0, but in order to uniformly define T we need to iterate along $\mathbf{N}$. (Formally writing down how this follows from ACA_0^+ gets a bit messy, but it essentially follows from the fact that we can regard the double sequence $T(n,m)$ as a set of natural numbers T' given by $((n,m),p) \in T'$ if and only if $p = T(n,m)$, and j is arithmetically recoverable from $(T')^j$.) We now define f by declaring $f(n,1) = T(n,1)$ and $f(n, m+1) = T(n, f(n+1, m))$ whenever $m > 0$. We will show that, for each $n \in \mathbf{N}$, $\langle f(n,m) \rangle_m$ satisfies the conclusions of the lemma for α_n. We do this by inducting on r. If $r = 1$, just put $y_1 = f(n,1)$ and take i to be such that $f(n,1) \in A_i$. Now suppose that $r > 1$ and suppose that the conclusions hold at $r-1$. Let $n \geq 1$ be arbitrary. Write $\langle w_j \rangle_{j=1}^{r-1}$ and $1 \leq k < p_n$ be given by the inductive hypothesis for α_{n+1} at $r-1$. Take i to be the unique number such that $k \in A_{n,i}$, where we use $A_{n,i}$ to denote the $i-$th set in the partition α_n. Put $y_1 = k$ and, for $1 < j \leq r$, put $y_j = \tau_n^{-1}(w_{j-1})$. Because the w_j's are separated, $\langle \tau^{-1}(w_j) \rangle$ is separated with respect to the basis $\langle w_j \rangle$, so applying τ_n to a finite sum of distinct y_j's gives a finite sum of distinct w_j's, whence the definition of α_{n+1} gives us that τ_n maps a finite sum of distinct w_j's to an element of $\tau_n(F_{\alpha_n}(k, p_n))$. As τ_n is injective, it follows that the sum of elements w_j's must be in $F_{\alpha_n}(k, p_n)$, whence we have that the sum is in $A_{n,i}$ and adding y_1 to the sum yields another element of $A_{n,i}$; as we chose y_1 to be in $A_{n,i}$, we have satisfied the first property. The second property follows (separation of $\langle y_m \rangle$ just follows from tracing back definitions and using the fact that each $\langle x_{n,m} \rangle_m$ is separated.

To verify the third property, note that $y_1 = k \leq p_n - 1 \leq f(n,1)$. Then, if $j > 1$, we have $w_{j-1} \leq f(n+1, j-1)$ by the inductive hypothesis, so $y_j \leq f(n,j)$ follows from applying τ_n^{-1} to both sides, as τ_n^{-1} is an increasing function. As we have shows that $f(n,m)$ satisfies the lemma for all α_n, it follows that $f(1,\cdot) = f_\alpha$ satisfies the conclusions for $\alpha_1 = \alpha$, as desired. (This proof is rather strange in that it uses induction to ultimately prove something about the base case!)

†

[4] This is not really a proof by contradiction, as we tried to show $P \implies Q$, so we assumed P, and clearly $(P \wedge Q) \implies Q$. The body of the proof showed $(P \wedge \sim Q) \implies Q$; together, they allow us to conclude $(P \wedge (Q \vee \sim Q)) \implies Q$, so $P \implies Q$.

Lemma 16. *The following is provable in* ACA_0^+. *If* $A_1, ..., A_q$ *forms a partition* α *of* $\mathbf{N}$, *there is a function* $f_\alpha : \mathbf{N} \to \mathbf{N}$ *and some* $1 \leq k \leq q$ *such that, for every* r, *there is some* $\langle y_j \rangle_{j=1}^r$ *such that* $FS(\langle y_j \rangle) \subset A_k$ *and* $y_j \leq f_\alpha(j)$ *whenever* $1 \leq j \leq r$.

Proof. Take f_α to be as in Lemma 15. For each $r \in \mathbf{N}$, write $\sigma(r)$ for the least i satisfying the conclusions of Lemma 15 (we will not use that it is the least i, but we need some method of choice). As the codomain of σ is $\{1, ..., q\}$, there must be some $k \in \{1, ..., q\}$ for which there are infinitely many r with $\sigma(r) = k$ (see our prior comments about cardinalities in second-order arithmetic). Now, for each r, we can choose some r' such that $r \leq r'$ and $\iota(r') = k$. Take $\langle y_j \rangle_{j=1}^r$ to be the restriction of the least finite sequence of length r' guaranteed by choice of r' and the fact that $\sigma(r') = k$ (here 'least' is with respect to some fixed encoding of finite sequences as natural numbers). One sees that k and $\langle y_j \rangle_{j=1}^r$ have the requisite properties directly from the conclusions of Lemma 15.

†

Now we are ready to prove Theorem 4 (Hindman's theorem) in ACA_0^+.

Proof. Write α for the partition $C_1, ..., C_k$. Take i, f_α as in 16. We now define a tree T by declaring that a finite sequence $\langle x_i \rangle_1^n$ is in T if and only if it is increasing, $FS(\langle x_i \rangle_1^n) \subset A_i$, and $x_j \leq f_\alpha(j)$ for all x_j. Note that T is finitely branching because, if $\langle x_i \rangle_1^n \in T$ and $\langle y_i \rangle_1^{n+1} \in T$ is an extension of $\langle x_i \rangle$, then $y_{n+1} \leq f_\alpha(n+1)$; clearly, there are only finitely may such $\langle y_i \rangle$. Now König's lemma implies that T has an infinite path, say $\langle x_m \rangle_1^\infty$. As any finite sum of distinct elements of $\langle x_m \rangle_1^\infty$ is a finite sum of distinct elements of $\langle x_m \rangle_1^n$ for some n and $FS(\langle x_m \rangle_1^n) \subset A_i$ since $\langle x_m \rangle_1^n \in T$, it follows that $FS(\langle x_m \rangle_1^\infty) \subset A_i$, as desired.

†

6 Current State of Affairs and Future Directions

The main current question about Hindman's theorem from a proof-theoretic standpoint is whether it is provable in ACA_0. Of course we are also interested in whether it proves ACA_0^+, but ACA_0 is a much more important subsystem of Z_2. At any rate showing that Hindman's theorem is equivalent to either one implies that it is not equivalent to the other. There are two primary heuristic reasons why it is unlikely that Hindman's theorem is equivalent to ACA_0. The first is that Carlucci recently showed that, if one considers the statement that $\mathrm{HT}_4^{\leq 3}$ implies ACA_0 (over the base system RCA_0). That is, the statement that, for every coloring of $\mathbf{N}$ with (at most) four colors, there is an infinite set and a particular color such that any sum of one, two, or three distinct elements of that set is of that fixed color, which appears to be much weaker than the full Hindman's theorem, is sufficient to prove ACA_0. So if Hindman's theorem were equivalent to ACA_0, it would be no stronger than this extreme restriction. The second reason why it is unlikely that Hindman's theorem is provable in ACA_0 is that ACA_0 has a minimal ω-model. If one considers the intersection of all collections of subsets of ω that form $\omega-$models of ACA_0, one obtains a collection of subsets of ω that forms an ω-model for ACA_0. In particular, the sets in this intersection are those computable from $\emptyset^{(n)}$ for some natural n. But we conjecture that Hindman's theorem does not have a minimal $\omega-$model. Let us consider how an $\omega-$model for Hindman's theorem might be constructed. We start out with the computable sets. Then, for each Hindman problem where the partition consists only of computable sets which does not have a computable solution, we pick some Hindman solution and adjoin it to our collection. We further enlarge our collection by saying that, if there are $S_1, ..., S_n$ in the collection such that $J \leq_T S_1 \oplus ... \oplus S_n$, we adjoin J as well. Now we can define more Hindman problems using the sets we just adjoined as well as the computable sets, and for each Hindman problem that does not have a solution among the sets we have already considered, we choose a solution and adjoin it, enlarging to include sets of lower Turing degree, etc. Iterating this process and taking the union of all of the collections we obtain clearly gives us an $\omega-$model for Hindman's theorem. But there were many arbitrary choices made. If it were the case that every Hindman problem has a solution that is a lower bound for all solutions to that problem (i.e. M is a solution such that, whenever N is a solution $M \leq_T N$), we could always choose such lower bounds when enlarging our collections and we would in fact obtain a minimal $\omega-$model for Hindman's theorem. But there is no reason why this should be the case (though it has not been proven that there is a Hindman problem with no such lower bound solution as far as I can tell). Different solutions can be wildly unrelated. In fact, it is not even clear whether each Hindman problem has a Turing-minimal solution (that is, a solution M such that there is no solution N with $N <_T M$). So such a construction will not go through, and if one makes different choices, there is no reason why the resulting collections of subsets should be related. We are not suggesting that the most promising avenue to proving that Hindman's theorem is not provable in ACA_0 is showing that it has no minimal $\omega-$model (though we are not suggesting that this is not the case, either). But it is a fairly strong heuristic reason why we should not expect Hindman's theorem to be provable in ACA_0.

Note that the question of the proof-theoretic strength of Hindman's theorem is not quite the same as the question of what the $\omega-$models of Hindman's theorem look like. It is perfectly possible that, say, all $\omega-$models of Hindman's theorem are models of ACA_0^+, but ACA_0^+ is not provable from Hindman's theorem. And even in terms of $\omega-$models, answering the question of which model Hindman's theorem is simpler than determining the minimal complexity of Hindman solutions; that is, if it were the case that, for every coloring, there is a Hindman solution computable in the second Turing jump of that coloring and there

is some coloring such that every solution computes the second Turing jump of that coloring, Hindman's theorem would have exactly the same $\omega-$models as ACA_0. But the same would be true if every coloring has a Hindman solution computable in, say, the fifth jump of that coloring and there were a coloring such that all Hindman solutions compute the fifth jump of that coloring, and for the same reasons. This is why we said earlier that the status of Hindman's theorem is murkier from a computability-theoretic perspective than from a proof-theoretic one. While ACA_0 and ACA_0^+ are closely related systems, we have essentially no knowledge of how complicated Hindman solutions become from a computability perspective. (Either of the aforementioned examples could be the case as far as we know.) At this stage, it remains likely that these questions about Hindman's theorem will be answered when a topological equivalent is found. Blass, Hirst, and Simpson show that Hindman's theorem is closely related to the Auslander-Ellis theorem in topological dynamics, and if some variant of the Auslander-Ellis theorem is obtained that can be shown to be proof-theoretically equivalent to Hindman's theorem, it seems likely that an exact placement of Hindman's theorem would follow shortly. Alternatively, one may note that we only used an axiom of ACA_0^+ that is not an axiom of ACA_0 at one point in the entire proof, and it is possible that this may be circumvented by a clever trick; such an argument could immediately yield a proof of Hindman's theorem in ACA_0.

References

[1] Andreas Blass, *Some Questions Arising from Hindman's Theorem*, http://www.math.lsa.umich.edu/∼ablass/hindman.pdf, 2004.

[2] Andreas Blass, Jeffry Hirst, Stephen Simpson, *"Logical Analysis of Some Theorems of Combinatorics and Topological Dynamics"*, Contemporary Mathematics: Logic and Combinatorics **65**, American Mathematical Society (1987), 125-156.

[3] Lorenzo Carlucci, *"Weak yet strong" restrictions of Hindman's Finite Sums Theorem*, Proceedings of the American Mathematical Society **146** (2018), 819-829.

[4] Barbara Csima, et.al., *The Reverse Mathematics of Hindman's Theorem for Sums of Exactly Two Elements*, Computability **8** (2019), 253-263.

[5] Martin Davis, *Computability & Unsolvability*, McGraw-Hill Book Company, New York, 1958.

[6] Neil Hindman, *Finite Sums from Sequences Within Cells of a Partition of N*, Journal of Combinatorial Theory **17** (1974), 1-11.

[7] ———, *The Existence of Certain Ultrafilters on N and a Conjecture of Graham and Rothschild*, Proceedings of the American Mathematical Society **36** (1972), 341-346.

[8] Denis Hirschfeldt, *Slicing the Truth: On the Computable and Reverse Mathematics of Combinatorial Principles*, World Scientific, Singapore, 2015.

[9] Antonio Montalbán, *Martin's Conjecture: A Classification of the Naturally Occurring Turing Degrees*, Notices of the American Mathematical Society **66** (2019), 1209-1215.

[10] ———, *Open Questions in Reverse Mathematics*, Bulletin of Symbolic Logic **17** (2011), 431-454.

[11] Hartley Rogers, Jr., *Theory of Recursive Functions and Effective Computability*, MIT Press, Cambridge, 1987.

[12] Stephen Simpson, *Subsystems of Second Order Arithmetic*, Cambridge University Press, New York, 2009.

[13] Rebecca Weber, *Computability Theory*, American Mathemtical Society, Student Mathematical Library **62**, 2012.