A Ramsey Theorem in Boyer-Moore Logic

Kenneth Kunen[1]
Computer Sciences Department
University of Wisconsin
Madison, WI 53706, U.S.A.
kunen@cs.wisc.edu

January 10, 1995

ABSTRACT

We use the Boyer-Moore Prover, Nqthm, to verify the Paris-Harrington version of Ramsey's Theorem. The proof we verify is a modification of the one given by Ketonen and Solovay. The Theorem is not provable in Peano Arithmetic, and one key step in the proof requires ϵ_0 induction.

§**0. Introduction.** The most well-known formalizations of finite mathematics are PA (Peano Arithmetic) and PRA (Primitive Recursive Arithmetic). In both, the "intended" domain of discourse is the set of natural numbers. PA is formalized in standard first-order logic, and contains the induction schema, which can apply to arbitrary first-order formulas. The logic of PRA allows only quantifier-free formulas, which are thought of as being universally quantified, and PRA has the induction scheme for quantifier-free formulas, expressed as a proof rule. Also, for each primitive recursive function f, PRA contains a function symbol for f and has the recursive definition of f as an axiom. Clearly, PRA is much weaker than PA. In particular, PA can prove the consistency of PRA by simply proving the statement "all sentences provable from PRA are true".

The Boyer-Moore theorem prover, Nqthm [2], is a Lisp-based system for computational logic. It allows the user to define functions recursively and to prove theorems about these functions. It is, to first approximation, an implementation of PRA, but it extends PRA in two important ways.

The first way is that it allows the use of symbolic expressions (Lisp S-expressions) as basic objects, as well as numbers. In theory, this extension is inessential, since these expressions can be encoded by Gödel numbering, but in practice, such a Gödel numbering would be extremely awkward to deal with. By dealing with S-expressions directly, Nqthm has become a practical verification tool, and has frequently been used to verify statements in finite combinatorics, as well as theorems about circuit design and algorithm correctness.

The second way *is* essential; Nqthm allows definition of functions by recursion on the ordinal ϵ_0, and proofs by induction on ϵ_0. Until now, the full strength of this extension has not been utilized. In this paper, we show how to utilize it.

In their book [2], Boyer and Moore point out that ordinals can be used to formalize double recursions. In many applications, this is only a matter of convenience, since the function defined is really primitive recursive, and could, with slightly more work, be defined

[1] This research was supported by NSF Grant DMS-9100665.

within PRA by a standard primitive recursion. The simplest example which goes beyond PRA is the Ackermann function, which grows faster than every primitive recursive function. Here, the recursion step is $f(x, y) = f(f(x-1, y), y-1)$ when $x, y > 0$. One can justify this double recursion as a simple recursion on the ordinal ω^2 by identifying (x, y) with the ordinal $\omega \cdot y + x$. The current Nqthm distribution contains a short Nqthm script, due to Kunen, which defines the Ackermann function and derives some of its properties.

However, ω^2 is a long way from ϵ_0. The earliest use of ϵ_0 in proof theory is due to Gentzen [3], who showed that in PRA plus ϵ_0 induction one can prove the consistency of PA. It might be an interesting exercise to implement this proof on Nqthm. However, in this paper, we choose instead a simple combinatorial theorem – namely, the Paris-Harrington version of Ramsey's Theorem.

Ramsey's Theorem states that one can, for each k, n, c, find a number $R(k, n, c)$ large enough so that whenever we partition all the n-tuples from the set $\{0, 1, 2, \ldots R(k, n, c)\}$ into c pieces, there is a set of size at least k which is *homogeneous* for the partition (that is, all of its n-tuples are in the same piece of the partition). The Paris-Harrington extension adds the seemingly harmless extra requirement that the homogeneous set be *large*, meaning that its size is at least as big as its first element. For example, $\{3, 5, 7\}$ is large but $\{4, 5, 7\}$ isn't.

Now, it is easy to see that Ramsey's Theorem can be proved within PRA, and the $n = 2$ case has been verified on Nqthm by Matt Kaufmann (see [1]). However, the Paris-Harrington extension cannot be proved even in full PA [5]. It can be proved in PRA plus ϵ_0 induction, as was demonstrated explicitly by Ketonen and Solovay [4], and hence, potentially, can be proved in Nqthm.

We have indeed verified this theorem on Nqthm, and describe the proof in this paper. In §1, we present the proof, using ordinary mathematical terminology. In §2, we explain how we formalized it on Nqthm. The complete Nqthm script is available by email from the author.

The reason we give the proof in §1, rather than just referring the readers to [4], is that the argument in [4] is a bit more complicated than it needs to be, and introduces a number of notions which turn out to be irrelevant. Our §1 may be of independent mathematical interest, since by simplifying the proof, we can pin down exactly where the ϵ_0 induction takes place. Many of the steps in [4] which appear to use induction on the ordinals can in fact be formalized in PRA.

The basic idea in the proof is the following: If X is a finite set of natural numbers, and $\alpha < \epsilon_0$, we define what it means for X to be α – *large*. For finite n, X is n – large iff $|X| = n$. Our "ω – large" is slightly stronger than "large", but as α grows above ω, the notion of α – large grows very rapidly; for example, if X is ω^2 – large and x is the first element of X, then $|X| > 2^{2^x}$. We then prove the *Ordinal Ramsey Theorem* (§1, Lemma 13): For each α, n, c, there is an ordinal $\Gamma(\alpha, n, c)$ such that whenever X is $\Gamma(\alpha, n, c)$ – large, every partition of the n – tuples from X into c pieces has an α – large homogeneous set. Actually, [4] proves this just for $\alpha = \omega$, which was sufficient to derive the Paris-Harrington Ramsey Theorem.

Now, the Ordinal Ramsey Theorem is done within PRA. Following Gentzen [3], and Boyer-Moore [2], each ordinal below ϵ_0 has a notation, which is a finite symbolic expression, and everything we need about ordinals is proved using ordinary induction on these notations. We only step beyond PRA in the proof that $\alpha-$large sets exist. Here, we define a number $\Lambda(\alpha, k) > k$ and prove that the set of numbers between k and $\Lambda(\alpha, k)$ is $\alpha-$large (§1, Lemma 14). $\Lambda(\alpha, k)$ is defined by *transfinite* recursion on $\alpha < \epsilon_0$. Our $R(k, n, c)$ is just $\Lambda(\Gamma(\omega, n, c), k)$.

It is easy to track the use of ϵ_0 in the Nqthm verification. Nqthm will not use ϵ_0 recursion in a function definition unless explicitly told to do so by the user, and Nqthm will not prove a theorem by ϵ_0 induction until at least one function has been defined by ϵ_0 recursion. The first (and only) use of ordinal recursion in our Nqthm code occurs in the definition of Λ, which is located after the proof of the Ordinal Ramsey Theorem.

§**1. The Proof.** For simplicity, we start by presenting the proof using ordinary set-theoretic notation. At the end of this section, we comment on formalizing it in PRA, and explain the relationship between our proof and the Ketonen-Solovay [4] proof.

Definition. A finite $X \subset \omega$ is called *large* iff X is non-empty and $|X| \geq \min(X)$.

Our goal is to prove:

Theorem. There is a computable function $R(k, n, c)$ such that: Whenever $c, n \geq 1$ and $\mathcal{F}$ is a partition of the n-tuples from $\{0, 1, 2, \ldots R(k, n, c)\}$ into c pieces, there is a large subset of $\{0, 1, 2, \ldots R(k, n, c)\}$ of size at least k which is homogeneous for $\mathcal{F}$.

For notational convenience, we identify ordinals, and in particular natural numbers, with von Neumann ordinals, so that each ordinal is the set of all smaller ordinals. Then each $c \in \omega$ is an c-element set. With this notation, if $\mathcal{F} : X \to c$ is a function in the usual mathematical sense, then $\mathcal{F}$ is a partition of X into c pieces, labeled $0, 1, \ldots, c-1$.

First, some ordinal arithmetic. We use $+$ and $\cdot$ to denote the usual ordinal addition and multiplication. As did Gentzen, we also make use of the the *natural sum*, $\alpha \# \beta$, which is obtained by merging the Cantor normal forms. More precisely, we define $0 \# \alpha = \alpha \# 0 = \alpha$. If α and β are both non-zero, we may express them as

$$\alpha = \sum_{j<r} \omega^{\delta_j} \cdot m_j \quad ; \quad \beta = \sum_{j<r} \omega^{\delta_j} \cdot n_j \quad ,$$

where $\delta_0 > \delta_1 > \cdots$ and the m_j, n_j are natural numbers, possibly 0. Then define

$$\alpha \# \beta = \sum_{j<r} \omega^{\delta_j} \cdot (m_j + n_j) \quad .$$

Note that $\#$ is associative and commutative. If α, β are non-0, then $\alpha \# \beta$ is larger than α and β. Also, $\alpha \# \beta$ is a successor ordinal iff at least one of α, β is a successor.

For finite n, $\alpha \star n$ is obtained by multiplying each coefficient in the Cantor normal form of α by n. Or, recursively, define $\alpha \star 0 = 0$ and $\alpha \star (n+1) = (\alpha \star n) \# \alpha$.

We say that (α, β) *mesh* iff each exponent occurring in the Cantor normal form of α is at least as large as every exponent occurring in the Cantor normal form of β. Note that if (α, β) mesh, then $\alpha \# \beta = \alpha + \beta$, and also (α, β') mesh for all $\beta' \leq \beta$.

If $\alpha < \epsilon_0$, then α may be written out as a finite symbolic expression, such as $\omega^{\omega^{3} \cdot 7} + \omega^9 \cdot 2 + 8$. We now define the *norm* of α, $\|\alpha\|$, which measures the complexity of this expression. Of course, there is a lot of choice in the technical details of the symbolism. Nqthm encodes Gentzen's notation for α into a Lisp S-expression, and our $\|\alpha\|$ is precisely the *count* of this notation (where "count" is the Nqthm measure of the complexity of S-expressions). By accident, this $\|\alpha\|$ is also precisely the definition of norm chosen by Ketonen and Solovay.

Definition. For $\alpha < \epsilon_0$, define the *norm* of α, $\|\alpha\|$, by: $\|0\| = 0$. If $0 < \alpha < \epsilon_0$, and α in Cantor normal form is

$$\alpha = \sum_{j<r} \omega^{\delta_j} \cdot m_j \quad ,$$

where $\alpha > \delta_0 > \delta_1 > \cdots$, and the m_j are positive natural numbers, then

$$\|\alpha\| = \sum_{j<r} (\|\delta_j\| + 1) \cdot m_j \quad ,$$

Let $E(n) = \{\alpha < \epsilon_0 : \|\alpha\| \leq n\}$.

Lemma 1.

1. $\|\alpha + 1\| = \|\alpha\| + 1$.
2. $\|n\| = n$ for $n \in \omega$.
3. $E(n)$ is finite.
4. $\|\alpha \# \beta\| = \|\alpha\| + \|\beta\|$.
5. $\|\omega^\alpha\| = \|\alpha\| + 1$.

Definition. Let $\{0\}(n) = 0$, and, for $0 < \alpha < \epsilon_0$, let $\{\alpha\}(n) = \max(\alpha \cap E(\|\alpha\| + 2n))$.

Note, by Lemma 1.2 and 1.3, that we are taking the max of a finite non-empty set here. The interest in the notion $\{\alpha\}(n)$ is that we have assigned to each limit α a "canonical" ω – sequence of ordinals converging up to α (Lemma 2.5). The rest of Lemma 2 summarizes some useful technical details.

Lemma 2. For $\alpha, \beta < \epsilon_0$ and $m, n < \omega$:

1. If $\alpha < \beta$ and $\|\alpha\| \leq \|\beta\| + 2n$, then $\alpha \leq \{\beta\}(n)$.
2. If α is a limit, then $\|\{\alpha\}(n)\| = \|\alpha\| + 2n$.
3. $\{\alpha + 1\}(n) = \alpha$.
4. If $m \leq n$ then $\{\alpha\}(m) \leq \{\alpha\}(n)$ and $\|\{\alpha\}(m)\| \leq \|\{\alpha\}(n)\|$.
5. If α is a limit, then the $\{\alpha\}(n)$ form a strictly increasing sequence of ordinals converging to α.
6. $\|\alpha\| - 1 \leq \|\{\alpha\}(n)\| \leq \|\alpha\| + 2n$
7. $\|\alpha \# \{\beta\}(n)\| \leq \|\alpha \# \beta\| + 2n \leq \|\{\alpha \# \beta\}(n)\| + 2n + 1$.
8. If $\beta > 0$, then $\{\alpha \# \beta\}(n) \geq \alpha \# \{\beta\}(n)$.
9. If (α, β) mesh and $\beta > 0$ then $\{\alpha + \beta\}(n) = \alpha + \{\beta\}(n)$.
10. $\{\omega\}(n) = 2n + 2$.

Proof. (1) – (5) are easy from the definitions and Lemma 1. (6) follows from (2) if α is a limit and from (3) if α is a successor. (7) follows from (6) and Lemma 1.4. For (8), $\{\beta\}(n) < \beta$ implies that $\alpha\#\{\beta\}(n) < \alpha\#\beta$, so by (7), $\alpha\#\{\beta\}(n) \in (\alpha\#\beta) \cap E(\|\alpha\#\beta\| + 2n)$; (8) now follows from the definition of $\{\alpha\#\beta\}(n)$. Now, (9) is immediate from (3) in the case that β is a successor, so assume that β is a limit. Since $\alpha < \alpha + \beta$ and $\|\alpha\| < \|\alpha + \beta\|$ (since + and # agree here), the definition of $\{\alpha + \beta\}(n)$ implies that $\{\alpha + \beta\}(n) \geq \alpha$, so we can set $\{\alpha + \beta\}(n) = \alpha + \beta'$. Applying (8), $\{\beta\}(n) \leq \beta' < \beta$. Applying (2) and Lemma 1.4, $\|\alpha\| + \|\beta'\| = \|\alpha\| + \|\beta\| + 2n$, so $\|\beta'\| = \|\beta\| + 2n$; but then, $\beta' = \{\beta\}(n)$ by the definition of $\{\beta\}(n)$. Finally, for (10), $\|\omega\| = \|\omega^1\| = \|1\| + 1 = 2$, so $\|\{\omega\}(n)\| = 2n + 2$ (by (2)), so $\{\omega\}(n) = 2n + 2$ by Lemma 1.2. ∎

We remark that the "$2n$" in the definition of $\{\alpha\}(n)$ could be replaced by any strictly increasing function of n, and all the basic results would hold unchanged. The choice of $2n$, rather than n, makes the detailed computations, especially Lemma 8, somewhat simpler. Actually, the $2n$ was a compromise between n and the function $(3n + 3) \cdot (n + 2)^{(n+2)}$, which would have made the Ramsey theorem simpler still, at the expense of making the elementary treatment of α – large sets look a bit ugly and artificial. This is discussed further at the end of this section.

Definition. If X is a finite subset of ω, and $\beta < \epsilon_0$, we define the notion "X is β – *large*" by: Every X is 0 – large. If $\beta > 0$, X is β – large iff X is non-empty and $X \backslash \{\min(X)\}$ is $\{\beta\}(\min(X))$ – large.

We shall usually prove theorems about large X by induction on $|X|$, but the following "sequence" approach may be useful for motivation. Let x_i, for $i < |X|$, list X in increasing order. Let $\beta_0 = \beta$ and $\beta_{i+1} = \{\beta_i\}(x_i)$ for $i < |X|$; so β_i is defined for $i \leq |X|$. Then X is β – large iff the last one, $\beta_{|X|}$, is 0. Note that the β_i^X decrease strictly as long as they are non-0. Informally, as β gets bigger, the sequence takes longer to reach 0, so that the notion of "β – large" gets large very rapidly as β increases; see Lemma 8 below.

Lemma 3. Let X be any finite subset of ω:
1. For $n < \omega$, X is n – large iff $|X| \geq n$.
2. X is ω – large iff X is non-empty and $|X| \geq 2\min(X) + 3$.
3. If X is ω – large then X is large.

Proof. (1) is by induction on n. For (2), use Lemma 2.10. Then, (3) is immediate from (2). ∎

Further results on the size of β – large sets are given in Lemma 8. Roughly, as β increases, the property "β – large" gets stronger, but this is not strictly true. For example, $\{1, 2, 3, 4, 5\}$ is ω – large but is not 6 – large. One can sometimes conclude from $\alpha \leq \beta$ plus X β – large that X is α – large, but one needs an additional assumption about the norm (see Lemma 4.3). It is always true that every subset of a β – large set is β – large (Lemma 4.2).

Lemma 4. Assume X is non-empty, X is β – large, and $x = \min(X)$.
1. If $X \subseteq Y$, $\alpha \leq \beta$, and $\|\alpha\| \leq \|\beta\| + 2x$, then Y is α – large.
2. If $X \subseteq Y$, then Y is β – large.
3. If $\alpha \leq \beta$, and $\|\alpha\| \leq \|\beta\| + 2x$, then X is α – large.

Proof. (2) and (3) are immediate from (1), which we now prove by induction on $|X|$. (1) is trivial if $\alpha = 0$, so assume $\alpha > 0$. Let $y = \min(Y)$. Then $y \le x$. We now consider three cases:

If $0 < \alpha = \beta$ and X is not a singleton: By Lemma 2.4 $\{\alpha\}(y) \le \{\alpha\}(x)$ and $\|\{\alpha\}(y)\| \le \|\{\alpha\}(x)\|$. Since $X \backslash \{x\}$ is $\{\alpha\}(x)$ – large, the Lemma, applied inductively to $(X\backslash\{x\}, Y\backslash\{y\})$, shows that $Y\backslash\{y\}$ is $\{\alpha\}(y)$ – large, so Y is α – large.

If $0 < \alpha = \beta$ and X is a singleton, then $\{\alpha\}(x) = 0$, so, by Lemma 2.4, $\{\alpha\}(y) = 0$, so Y is α – large.

If $0 < \alpha < \beta$, then by Lemma 2.1, $\alpha \le \{\beta\}(x)$. Now, $X\backslash\{x\}$ is $\{\beta\}(x)$ – large, and in particular non-empty, so let $x_1 = \min(X\backslash\{x\})$. Then $x_1 \ge x + 1$. Applying Lemma 2.6,

$$\|\alpha\| \le \|\beta\| + 2x \le \|\{\beta\}(x)\| + 1 + 2x \le \|\{\beta\}(x)\| + 2x_1 \quad .$$

So, the Lemma, applied inductively to $(X\backslash\{x\}, Y)$, shows that Y is α – large. ■

Lemma 5. If $X = A \cup B$ and X is $\alpha\#\beta$ – large, then either A is α – large or B is β – large.

Proof. Induct on $|X|$. Assume $\alpha, \beta > 0$, and X has at least two elements; otherwise, the result is trivial. Let x_0, x_1 be the first two elements of X. By symmetry, we may assume that x_0 is in B; it may or may not be in A. Note that $X\backslash\{x_0\}$ is $\{\alpha\#\beta\}(x_0)$ – large. Applying Lemma 2, $\{\alpha\#\beta\}(x_0) \ge \alpha\#\{\beta\}(x_0)$, and

$$\|\alpha\#\{\beta\}(x_0)\| \le \|\alpha\#\beta\| + 2x_0 \le \|\{\alpha\#\beta\}(x_0)\| + 2x_0 + 1 \le \|\{\alpha\#\beta\}(x_0)\| + 2x_1 \quad ,$$

so by Lemma 4.3, $X\backslash\{x_0\}$ is $\alpha\#\{\beta\}(x_0)$ – large. Applying the Lemma inductively to $X\backslash\{x_0\}$, either $A\backslash\{x_0\}$ is α – large, whence A is α – large (by Lemma 4.2), or $B\backslash\{x_0\}$ is $\{\beta\}(x_0)$ – large, whence B is β – large. ■

Lemma 6. If X is $\alpha \star c$ – large, $c \ge 1$, and $X = \bigcup_{i<c} A_i$, then some A_i is α – large.

Proof. Induct on c, using Lemma 5. ■

Lemma 6 is a Ramsey theorem for 1-tuples. To prove a general Ramsey theorem, we first (Lemma 8) need to get some rough lower bound estimates on the size of α-large sets.

Definition. X is *below* Y iff $x < y$ for all $x \in X$ and all $y \in Y$.

Note that this notion is vacuously true if either X or Y is empty.

Lemma 7. If (α, β) mesh and X is $\alpha + \beta$ – large, then X can be partitioned into disjoint sets P, Q such that P is below Q, P is β-large, and Q is α-large.

Proof. Induct on $|X|$. Assume X is non-empty and $\beta > 0$, since otherwise the result is trivial. Now, apply the lemma inductively to $X\backslash\{\min(X)\}$, using Lemma 2.9. ■

Lemma 8. Let X be any finite non-empty subset of ω and $x = \min(X)$.

1. For $0 < n < \omega$, if X is $\omega \cdot n$ – large then $|X| \ge 2^n(x+1)$.
2. If X is ω^2 – large then $|X| \ge 2^{x+1}(x+1)$.
3. If X is $\omega^2 \cdot 2$ – large then
$$|X| \ge 2^{2^{x+1}} \cdot 2^{x+1} \quad .$$

4. If X is $\omega^2 \cdot 2 + n$ – large then

$$|X| \geq (n)^{2^x} \cdot (n + 2x + 4) \quad .$$

Proof. For (1), we induct on n; the case $n = 1$ follows from Lemma 3. Assuming (1) for n, let X be $\omega \cdot (n+1)$ – large. Apply Lemma 7 with $\alpha = \omega \cdot n$ and $\beta = \omega$. Let $q = \min(Q)$; Q is $\omega \cdot n$ – large, so $|Q| \geq 2^n(q+1)$. P is ω – large, so $|P| \geq 2x + 3$, so $q \geq x + |P| \geq 3x + 3$. So, $|Q| \geq 2^n(3x+4) > 2^{n+1}(x+1)$.

For (2), $\|\omega^2\| = 3$ and $\|\omega \cdot (x+1)\| = 2x + 2 < 3 + 2x$, so by Lemma 4.3, if X is ω^2 – large, it is also $\omega \cdot (x+1)$ – large. Now, (2) follows from (1) with $n = x + 1$.

For (3), apply Lemma 7 with $\alpha = \beta = \omega^2$. Let $q = \min(Q)$. (2) for Q implies that $|Q| \geq 2^q \cdot q$, and (2) for P implies that $q \geq |P| \geq 2^{x+1}$. Now, (3) follows immediately. Using (3) and the same method, if X is $\omega^2 \cdot 2 + n$ – large, then

$$|X| \geq 2^{2^{x+n+1}} \cdot 2^{x+n+1} \quad ,$$

and (4) now follows by elementary arithmetic. ∎

Items (1) – (3) of Lemma 8 give some idea of how large α – large sets get. Item (4) is completely uninteresting, but turns out to be just what is needed for Lemma 11 below. If X is ω^ω – large, $|X|$ must be at least something like the Ackermann function of x. One might now ask whether one can even produce α – large sets for all $\alpha < \epsilon_0$. This is done in Lemma 14 below. First, we show how α – large sets are used in the Ramsey theorem.

Definition. Let $\varphi(\alpha, c) = \omega^\alpha + \omega^2 \cdot 2 + \|\alpha\| + c$. Let $\Gamma(\alpha, n, c)$ be $\alpha \star c$ if $n \leq 1$ and $\varphi(\Gamma(\alpha, n-1, c), c)$ if $n \geq 2$.

Definition. $[\omega]^{<\omega}$ is the family of all finite subsets of ω. If $\mathcal{F} : [\omega]^{<\omega} \to c$, we say that $HOM(V, \mathcal{F}, n)$ holds iff for all $S, T \subseteq V$, if $|S| = |T| = n$, then $\mathcal{F}(S) = \mathcal{F}(T)$. Let $PH(Z, \mathcal{F})$ abbreviate the statement that for all non-empty $S \subseteq Z$ and all $a, b \in Z$ such that $a, b > \max(S)$, $\mathcal{F}(S \cup \{a\}) = \mathcal{F}(S \cup \{b\})$.

Some remarks: The discussion of Ramsey's Theorem is technically somewhat simpler if we consider partitions to act on all finite subsets of ω, although the conclusion of the Theorem involves only the partition being constant on n-tuples from a given set. The basic Theorem is really Lemma 13, below – that is, if X is $\Gamma(\alpha, n, c)$ – large and the n-tuples from X are partitioned into c pieces, then there is an α – large homogeneous $V \subseteq X$. The case $n = 1$ of this was Lemma 6. So, we proceed by induction. In one of the standard inductive proofs of Ramsey's Theorem, when we wish to obtain a V with $HOM(V, \mathcal{F}, n+1)$, we first find a big $Z \subseteq X$ which is *pre-homogeneous* ($PH(Z, \mathcal{F})$) – that is, for $S \subseteq Z$, $\mathcal{F}(S)$ doesn't depend on the last element of S, so that the action of $\mathcal{F}$ on the $n+1$-tuples from Z can be reduced to a partition on n-tuples. Formally,

Definition. If $\mathcal{F} : [\omega]^{<\omega} \to c$ and $k \in \omega$, then the *derived partition*, $\partial_k(\mathcal{F})$, is the $\mathcal{G} : [\omega]^{<\omega} \to c$ defined by: $\mathcal{G}(S) = \mathcal{F}(S \cup \{k\})$.

This notion will be used only in the context of the following Lemma:

Lemma 9. If $\mathcal{F} : [\omega]^{<\omega} \to c$, $\emptyset \neq Z \subseteq \omega$, $PH(Z, \mathcal{F})$, $n \geq 1$, $k = \max(Z)$, $V \subseteq Z$, and $HOM(V, \partial_k(\mathcal{F}), n)$, then $HOM(V, \mathcal{F}, n+1)$

Thus, we shall get a homogeneous set for $n+1$-tuples by applying Ramsey's Theorem for n-tuples to the derived partition on a pre-homogeneous subset, which is obtained using Lemma 12 below, which says that if X is $\varphi(\alpha, c)$ – large, then X has an α – large pre-homogeneous subset. The pre-homogeneous subset is in turn extracted by induction, but the induction is quite a bit simpler if we obtain a stronger property, "very pre-homogeneous".

Definition. Let $\mathcal{F} : [\omega]^{<\omega} \to c$. $VPH(Z, \mathcal{F})$ abbreviates the statement that for all non-empty S and all $a, b \in Z$, if $\max(S) \in Z$ and $a, b > \max(S)$, then $\mathcal{F}(S \cup \{a\}) = \mathcal{F}(S \cup \{b\})$. If $x \in \omega$, $NICE(x, Z, \mathcal{F})$ abbreviates the statement that for all $a, b \in Z$, and all (possibly empty) $S \subseteq x$, $\mathcal{F}(S \cup \{x, a\}) = \mathcal{F}(S \cup \{x, b\})$.

Lemma 10.
1. $VPH(Z, \mathcal{F})$ implies $PH(Z, \mathcal{F})$.
2. If $VPH(Z, \mathcal{F})$, $x < \min(Z)$, and $NICE(x, Z, \mathcal{F})$, then $VPH(\{x\} \cup Z, \mathcal{F})$.

Note that $VPH(Z, \mathcal{F})$ is true trivially if $|Z| < 3$. By applying Lemma 10.2 repeatedly, we can build up arbitrarily large pre-homogeneous sets if we can get the "nice" hypothesis to hold, which we do in the next lemma.

Lemma 11. Assume $\alpha \geq 3$ and $c \geq 1$. Let $\mathcal{F} : [\omega]^{<\omega} \to c$, and suppose that X is $\varphi(\alpha, c)$ – large. Let $x = \min(X)$ and $\hat{\alpha} = \{\alpha\}(x)$. Then there is an $N \subseteq X \backslash \{x\}$ such that N is $\varphi(\hat{\alpha}, c)$ – large and $NICE(x, N, \mathcal{F})$.

Proof. Let $d = \|\alpha\| + c$, so that X is $(\omega^\alpha + \omega^2 \cdot 2 + d)$ – large. Since $\alpha \geq 2$, $(\omega^\alpha, \omega^2 \cdot 2 + d)$ mesh, so by Lemma 7, we can partition X into P, Q, where P is below Q, P is $(\omega^2 \cdot 2 + d)$ – large and Q is ω^α – large. Let $q = \min(Q)$. Applying Lemma 8.4 $q \geq |P| \geq (d + 2x + 4) \cdot d^{2^x}$.

Call $a, b \in Q$ *equivalent* iff $\mathcal{F}(T \cup \{x, a\}) = \mathcal{F}(T \cup \{x, b\})$ whenever $T \subseteq x$. Note that for $N \subseteq Q$, $NICE(x, N, \mathcal{F})$ will hold iff all elements of N are equivalent. So, we shall choose $N \subseteq Q$ to be some equivalence class. Note that there are at most c^{2^x} equivalence classes on Q. Thus, by Lemma 6, there is a $\varphi(\hat{\alpha}, c)$ – large equivalence class if Q is $\varphi(\hat{\alpha}, c) \star c^{2^x}$ – large. Now, we know that Q is ω^α – large. Since $\alpha \geq 3$, $\omega^\alpha > \varphi(\hat{\alpha}, c) \star c^{2^x}$. Also,

$$\|\varphi(\hat{\alpha}, c) \star c^{2^x}\| \leq (\|\hat{\alpha}\| + 1 + 6 + \|\hat{\alpha}\| + c) \cdot c^{2^x} \leq (2\|\alpha\| + 4x + 7 + c) \cdot c^{2^x} \leq (2d + 4x + 8) \cdot d^{2^x} \leq 2q \ ,$$

so Q is indeed $\varphi(\hat{\alpha}, c) \star c^{2^x}$ – large by Lemma 4.3. ■

Lemma 12. Let $c \geq 1$, $\mathcal{F} : [\omega]^{<\omega} \to c$, and suppose that X is $\varphi(\alpha, c)$ – large. Then there is a $Z \subseteq X$ such that Z is α – large and $VPH(Z, \mathcal{F})$.

Proof. We induct on $|X|$. Assume $\alpha \geq 3$, since otherwise we may take Z to be any α-element subset of X. Applying Lemma 11, let $x = \min(X)$, let $\hat{\alpha} = \{\alpha\}(x)$, and let $N \subseteq X \backslash \{x\}$ be such that N is $\varphi(\hat{\alpha}, c)$ – large and $NICE(x, N, \mathcal{F})$. $|N| < |X|$, so by induction, we may choose $\hat{Z} \subseteq N$ such that $VPH(\hat{Z}, \mathcal{F})$ and $\hat{Z}$ is $\{\alpha\}(x)$ – large. Note that $NICE(x, \hat{Z}, \mathcal{F})$. Let $Z = \hat{Z} \cup \{x\}$. Then Z is α – large (by the definition of α – large), and $VPH(Z, \mathcal{F})$ (by Lemma 10.2 applied to $\hat{Z}$). ∎

Lemma 13 (*Ordinal Ramsey Theorem*)**.** Assume $c, n \geq 1$, $\mathcal{F} : [\omega]^{<\omega} \to c$, and X is $\Gamma(\alpha, n, c)$ – large. Then there is a $V \subseteq X$ such that $HOM(V, \mathcal{F}, n)$ and V is α – large.

Proof. For $n = 1$, this is Lemma 6. Applying induction, assume the Lemma holds for n, and assume that X is $\Gamma(\alpha, n+1, c)$ – large. Since $\Gamma(\alpha, n+1, c) = \varphi(\Gamma(\alpha, n, c), c)$, we may use Lemma 12 to fix $Z \subseteq X$ such that Z is $\Gamma(\alpha, n, c)$ – large and $VPH(Z, \mathcal{F})$, and hence $PH(Z, \mathcal{F})$. Let $k = \max(Z)$. Applying the Lemma to the derived partition $\partial_k(\mathcal{F})$, we get an α – large $V \subseteq Z$ such that $HOM(V, \partial_k(\mathcal{F}), n)$, and hence, by Lemma 9, $HOM(V, \mathcal{F}, n+1)$. ∎

The main Theorem is really a special case of this (with $\alpha = \omega$), except that we still need to verify that a $\Gamma(\alpha, n, c)$ – large X really exists. This is easy, using ϵ_0 induction:

Definition. $\Lambda(0, k) = k$. If $0 < \alpha < \epsilon_0$, $\Lambda(\alpha, k) = \Lambda(\{\alpha\}(k), k+1)$. Let $R(k, n, c) = \Lambda(\Gamma(\omega, n, c), k)$.

Lemma 14. For each $\alpha < \epsilon_0$ and $k < \omega$: $k \leq \Lambda(\alpha, k)$, and the interval $[k, \Lambda(\alpha, k)]$ is α – large.

Proof. Induct on α. ∎

Proof of Theorem. This is immediate from Lemmas 13 and 14, plus the fact that every ω – large set is large (Lemma 3.3). ∎

We comment further on the relationship between our proof and the one in Ketonen-Solovay [4]. As noted above, our definition of $\|\alpha\|$, is exactly the same as in [4] and in Nqthm. The symbolism $\{\alpha\}(n)$ is taken from [4], but we give it a slightly different value. The key fact about $\{\alpha\}(n)$ (Lemma 2.5) is that for limit α, we have chosen a "canonical" ω – sequence of ordinals converging up to α. From a purely set-theoretic point of view, the definition in [4] seems more natural than ours. For example, their $\{\omega\}(n)$ is n, whereas ours is $2n+2$. However, computations involving both $\{\alpha\}(n)$ and norms form the bulk of the proof, and the definition in [4] makes these computations rather complicated, whereas if we define $\{\alpha\}(n)$ the way we do, in terms of the norm, the computations are very simple. Any Ramsey theorem will eventually rely on some pigeonhole principle. Ours, Lemmas 5 and 6, is exactly what you would expect it to be, whereas there is no simple analogous result in [4].

The step of obtaining a pre-homogeneous set is a standard one in Ramsey theory, but once we have a simple pigeonhole principle, it is easy to produce the pre-homogeneous set by a simple induction, rather than by a tree argument as in [4].

Everything in our proof through Lemma 13 can be formalized within PRA. This includes, in particular, the definition of Γ and the fact that it works in a Ramsey theorem. It is only when we need the existence of α – large sets and the functions Λ and R that we have to apply ϵ_0 induction. Up to this point, all of our theorems about finite sets X were done by simple induction on the size of X. Besides that, we used a certain amount of ordinal arithmetic, dealing with ordinal sum, product, and exponentiation, and the order on the ordinals. However, if we view this arithmetic as expressing statements about the notations for ordinals, it can all be done in PRA. Several of the results were stated as existential facts, but these can all be re-stated in purely universal form, by defining a primitive recursive function. For example, Lemma 13 says that "there is" a homogeneous set V; when formalized in PRA, the proof defines a primitive recursive function which returns such a V.

As we remarked after Lemma 2, if $\mu : \omega \to \omega$ is any strictly increasing function, then we may replace "$2n$" by "$\mu(n)$" everywhere in the theory. In particular, if $0 < \beta < \epsilon_0$: define $\{\beta\}(n, \mu) = \max(\beta \cap E(\|\beta\| + \mu(n)))$, and say that X is (β, μ) – large iff X is non-empty and $X \backslash \{\min(X)\}$ is $(\{\beta\}(\min(X)), \mu)$ – large. Now Lemma 2.6 says that $\|\alpha\| - 1 \leq \|\{\alpha\}(n, \mu)\| \leq \|\alpha\| + \mu(n)$, and Lemma 4.3 says that if X is (β, μ) – large, $\alpha \leq \beta$, and $\|\alpha\| \leq \|\beta\| + \mu(\min(X))$, then X is (α, μ) – large.

Let $\mathcal{I}$ be the identity function on ω. Note that X is (β, μ) – large iff $\mu(X)$ is $(\beta, \mathcal{I})$ – large. So, considering various μ does not add anything essential to the theory, although our choice of $\mu(n) = 2n$ simplifies the arithmetic somewhat.

By a rather artificial choice of μ, we may omit a number of ordinal computations in the proof. This is primarily of interest for the Nqthm verification. Humans would probably prefer a simple and natural μ. Nqthm doesn't care about naturalness, and is optimized to verify complicated facts about arithmetic on the integers, but has to struggle with "obvious" facts about ordinal arithmetic.

In particular, all we really need from Lemma 8 is the fact that if X is $(\omega + d, \mu)$ – large then $|X| \geq \mu(\min(X) + d)$, and we can replace the definition of φ by $\varphi(\alpha, c) = \omega^\alpha + \omega + \|\alpha\| + c$. We do not really need the concept of "mesh", and all we need from Lemma 7 is the fact that if Z is $(\omega^\alpha + \omega + d, \mu)$ – large, then there is an (ω^α, μ) – large $Q \subseteq Z \backslash \{\min(Z)\}$ with $\min(Q) \geq \mu(\min(Z) + d)$.

To find the appropriate μ for this to work we examine the proof of Lemma 11 and see what μ we need, namely $\mu(s) = (3s + 3) \cdot (s + 2)^{(s+2)}$. In the proof of Lemma 11, let $d = \|\alpha\| + c$ and $s = z + d$. So, Z is $(\omega^\alpha + \omega + d, \mu)$ – large. With Q as above, we have that Q is (ω^α, μ) – large, and we need that Q is $(\varphi(\hat{\alpha}, c) \star c^{2^z}, \mu)$ – large. Our estimates are now:

$$\|\varphi(\hat{\alpha}, c) \star c^{2^z}\| \leq (\|\hat{\alpha}\| + 1 + 2 + \|\hat{\alpha}\| + c) \cdot c^{2^z} \leq (2\|\alpha\| + 2\mu(z) + 3 + c) \cdot c^{2^z} \leq$$
$$(2s + 3 + 2\mu(s)) \cdot s^{2^s} \leq 3\mu(s) \cdot s^{2^s} \leq \mu(\mu(s)) \leq \mu(\min(Q)) \quad ,$$

so Q is indeed $(\varphi(\hat{\alpha}, c) \star c^{2^z})$ – large by the revised Lemma 4.3.

Of course, μ was simply chosen so that these inequalities would work out. Besides μ being a strictly increasing function, μ was cooked up so that:

$$\mu(s) \geq 2s + 3$$
$$\mu(r) \geq 3r \cdot r^r$$
$$\mu(s) \geq 2^s$$
$$3\mu(s) \cdot s^{2^s} \leq 3\mu(s) \cdot \mu(s)^{\mu(s)} \leq \mu(\mu(s))$$

§**2. The Implementation.** The material in §1 was implemented and verified on the Boyer-Moore prover, Nqthm. The script comprises about 280 Kbytes of code, and takes a standard workstation between 15 and 30 minutes to verify – a lot quicker than most humans would be able to digest §1. The script ends with the main Theorem, stated as:

```
(prove-lemma ramsey-P-H () (implies
  (and
    (rangep g c)          ; g maps into 0 ...  c-1
    (not (zerop n))       ; we are partitioning n-tuples for some n > 0
    (numberp k)           ; k = the desired size of the homogeneous set
    (equal R (R k n c)) ; the Ramsey number
    (equal YY (extract-ramsey-P-H g k n c)) )
           ; YY is the computed homogeneous set
  (and
    (setp YY)
        ; YY is a set
    (subsetp YY (segment 0 R))
        ; YY is a subset of 0, 1,  ...  R
    (homp YY g n)
        ; YY is homogeneous for g as a partition of n-tuples
    (leq k (length YY))
        ; YY has size at least k
    (largep YY) )
        ; YY is large in the Paris - Harrington sense
))
```

It is "obvious" from the comments that this expresses the Paris-Harrington Ramsey Theorem stated in §1. However, as usual in the expression of a mathematical theorem within a formal system, one must first address the issue of whether the formal statement corresponds to the informal intent of the theorem. For example, although the comment states that `(homp YY g n)` means that `YY` is homogeneous, the actual Nqthm definition is a bit complex, relying on a sequence of preliminary definitions. Perhaps the author, either by design or through stupidity, entered a definition which does not correctly capture the notion of "homogeneous", perhaps making the whole theorem trivial. As pointed out in [2], since this issue connects the formal and informal notions, it can never be settled formally, but we shall make a few remarks in an effort to convince the reader that our definitions are correct. We take up the notions appearing in the Theorem in order of increasing difficulty.

First, the following notions are built-in to Nqthm: `implies`, `and`, `not` and `equal` have the obvious meanings. `(numberp k)` means that `k` is a natural number. All Nqthm functions (either built-in or defined by the user) are total; in particular, built-in numeric functions cast non-numeric input to the number 0. Thus, `(zerop n)` means that either `n` is a non-number or `n = 0`, so that `(not (zerop n))` means that `n` is a positive natural number. `leq` means $\leq$.

The function `length` is not built-in, but we defined it in the obvious recursive way:

```
(defn length (lst) (if (listp lst)
    (add1 (length (cdr lst)))
    0 ))
```

This is exactly as suggested in [2].

We have found it convenient to represent sets as increasing lists of natural numbers, so that each set of numbers has a unique representation. Then `(setp YY)` says that "`YY` is a standard representation of a set". This is defined by:

```
(defn setp (s) (if (listp s)
    (and
        (numberp (car s))
        (setp (cdr s))
        (or
            (equal (cdr s) nil)
            (lessp (car s) (cadr s)) ))
    (equal s nil)  ))
```

So, for example, `(setp NIL)` and `(setp '(1 4 5))` evaluate to `T` (*true*), whereas `(setp '(1 6 5))` evaluates to `F` (*false*). Also, `(subsetp s1 s2)` is defined by recursion on `s1` to say that every member of `s1` is a member of `s2`;

```
(defn subsetp (s1 s2) (if (listp s1)
        (and
            (member (car s1) s2)
            (subsetp (cdr s1) s2))
        T ))
```

Since each set has a unique representation, extensionality holds; that is, we establish:

```
(prove-lemma extensionality ( ) (implies
    (and (setp s1) (setp s2) (subsetp s1 s2) (subsetp s2 s1))
    (equal s1 s2) ))
```

If `YY` represents a set, then `(length YY)` is just the size of the set. It is now clear how to define `(largep YY)`:

```
(defn largep (set) (and
    (setp set) (listp set)
    (not (lessp (length set) (car set)))))
```

The function `(segment m n)` was defined so that for `m` $\leq$ `n`, it would return the list of numbers from `m` to `n` in increasing order, which is our representation of the set of numbers from `m` to `n`:

```
(defn segment (m n) (if
    (and (leq m n) (numberp m) (numberp n))
    (cons m (segment (add1 m) n))
     nil  )
( ; hint
(lessp (difference  (add1 n) m))
))
```

The `hint` is needed here because the recursion goes upward on `m`.

Now, the Theorem quantifies over arbitrary partitions, `g`. A partition is a special kind of function, and we follow the standard Lisp convention of representing (finite) functions as lists of ordered pairs, called *association lists*. Then, `(funcall g x)` is intended to be the value of the function `g` on the argument `x`. Since the function `assoc` is built-in to Nqthm, we defined `funcall` simply as

```
(defn funcall (g x) (cadr (assoc x g)))
```

For example,

```
( (1 5) (2 8) (3 0) )
```

represents the function which takes 1 to 5, 2 to 8, and 3 to 0. If `x` is not in the "intended" domain of `g` ($\{1, 2, 3\}$ in this example), then `(funcall g x)` returns 0 (since in Nqthm, unlike in Common Lisp, the `car` and `cdr` of a non-list is 0).

It is clear that in this way, we can represent any finite function. We now define `(rangep g c)` to say `c` is a positive number and that every value, `(funcall g x)`, is a number less than `c`. Since 0 is the default function value, `rangep` can be defined simply by recursion on the list `g`:

```
(defn rangep (g c) (if (nlistp g)
    (lessp 0 c)
    (and
        (rangep (cdr g) c)
        (numberp (cadar g))
        (lessp (cadar g) c))))
```

As in §1, it is technically simpler to think of partitions of n-tuples into c pieces as functions which map *everything* into $\{0, 1, \ldots, c-1\}$. Thus, the hypotheses to our statement of Ramsey's Theorem do not say anything about `g` partitioning n-tuples.

The n-tuples occur only inthe notion of *homogeneous*, `(homp YY g n)`, which says that g has the same value on all n-tuples from YY. There is no simple way to define this by recursion. We first defined `(power-set YY)` to return a list containing all subsets of `YY`, and then defined `(product lst1 lst2)` to return a list containing all pairs `(x . y)` with `x` in `lst1` and `y` in `lst2`. Then we defined

```
(defn homp (YY g n) (nlistp (counter-hom YY g n
    (product (power-set YY) (power-set YY))  )))
```

where the function `(counter-hom YY g n list-of-pairs)` runs down `list-of-pairs` and tries to find (and return) a pair of sets `(S1 . S2)` on the list such that `S1,S2` have size `n` and `(funcall g S1)` differs from `(funcall g S2)`. If it fails, then it returns 0. So, `(homp YY g n)` says that there is no counter-example to homogeneity on the list of all pairs of `n`-element subsets of `YY`. Since this is all a bit complicated, there is a danger of a programming error here. To verify the correctness of our definition, we proved a lemma stating that `(homp YY g n)` indeed implies that `g` is constant on all `n`-tuples from YY:

```
(prove-lemma homp-is-sufficient (rewrite) (implies
    (and
        (homp YY g n)
        (subsetp x YY) (subsetp y YY)
        (setp YY) (setp x) (setp y)
        (equal (length x) n) (equal (length y) n) )
    (equal (funcall g x) (funcall g y))    ) )
```

That concludes our justification of the *statement* of the theorem. Note that we are not required to justify our definitions of the function `(R k n c)`, which computes the Ramsey number, and the function `(extract-ramsey-P-H g k n c)`, which extracts the homogeneous set for the Paris-Harrington form of Ramsey's Theorem. We claim that they were defined by implementing the discussion in §1, but even if we are lying here, and have computed them in some completely different way, we have still proved the theorem.

In fact, the Paris-Harrington Ramsey Theorem is an easy corollary of the Ordinal Ramsey Theorem (§1, Lemma 13), which we consider to be the main result. This is stated as:

```
(prove-lemma ord-ramsey (rewrite) (implies
    (and
        (ordinalp alpha)
        (rangep g c)
        (not (zerop n))
        (setp ZZ)
        (o-largep ZZ (Gamma alpha n c))
        (equal YY (extract-ramsey  g ZZ alpha n c))  )
    (and
        (setp YY)
        (subsetp YY ZZ)
        (o-largep YY alpha)
        (homp YY g n))      ))
```

Here, `(o-largep YY alpha)` expresses the notion that the set `YY` is `alpha` – large, and the function `extract-ramsey` extracts the homogeneous set for the Ordinal Ramsey Theorem.

We now make some further remarks on our Nqthm script, as a guide to those readers who may wish to look at it in detail. Roughly, the script follows the outline of the human-readable proof presented in §1. However, since computers and humans differ in what they find difficult, we departed from this outline somewhat. In particular, §1 took for granted some elementary facts about ordinals, whereas Nqthm contains nothing about ordinals beyond the definitions of `ordinalp` and `ord-lessp`, and the ability to define functions by recursion on `ord-lessp`. This contrasts with the situation for natural numbers, where the prover not only contains some basic functions (such as $+$ and $*$), but also has a built-in decision procedure for linear arithmetic. Thus, as a practical matter, it is much easier to do arithmetic on ω than on ϵ_0. So, in the implementation, we took the tack described at the end of §1, using the function $\mu(s) = (3s+3) \cdot (s+2)^{(s+2)}$, which we called `(magic s)`. In this way, we avoided having to define ordinal sum and product, and the concept of "mesh", although we did need to define $\alpha\#\beta$. Throughout the script, the notion of α – large actually means (α, μ) – large in the notation of §1.

Our script begins with a section on basic arithmetic. This section verifies all the facts about the function `magic` which get used later in the proof, such as the inequalities at the end of §1. A human might find all this tedious and uninteresting, but the computer finds it trivial. This section also contains some other simple arithmetic facts, such as the associativity of multiplication.

We then proceeded to verify some basic facts about lists, and about our representation of sets of numbers by lists. Two different notions of "sublist" suggest themselves here. One, `(subsetp s1 s2)`, said that every member of `s1` is a member of `s2`. The other, `(sublistp s1 s2)`, said that `s1` can be obtained by deleting 0 or more elements of `s2` (but not changing the order). We proved these notions to be equivalent on our "standard" representations for sets of numbers.

We then took up ordinals. It was easy to verify the basic facts about successor and limit ordinals and the fact that `ord-lessp` is a total order. The definition and properties of $\alpha\#\beta$ took more work. As in §1, this is defined by merging the Cantor normal forms, which are easily read off of the Nqthm representation of the ordinals, so we defined `(sharp alpha beta)` roughly the way one would recursively define a `merge` in Lisp. However, we needed a special case for the coefficient of ω^0, which is treated differently from the coefficients of the other ω^ξ in Nqthm; this made all the proofs a bit complicated. We verify here that $\alpha\#\beta$ is commutative and associative, and increasing in each argument.

Next come the notions of norm ($\|\alpha\|$) and $\{\alpha\}(n)$. Now, `(norm x)` was simply defined to be `(count x)`; we only made this definition so that later a `(disable norm)` would not disable `count`. The definition of $\{\alpha\}(n)$ as $\max(\alpha \cap E(\|\alpha\| + \mu(n)))$ required first defining a function which represented each $E(k)$ as a list. We then verified Lemmas 1 and 2, defined the notion of $\alpha - large$ (`(o-largep YY alpha)`), and verified Lemmas 3 and 4.

Lemma 5, the "two-set pigeon-hole principle" was proved in §1 in eight lines, but it took about 750 lines of Nqthm code to verify it; the proof in §1 used without mention some trivial facts about finite sets, none of which are known to Nqthm. The c-set pigeon-hole principle, Lemma 6, took about 700 more lines to verify. Although the proof of Lemma 6 is just one line in §1, saying, essentially, "the obvious induction works", Nqthm must switch gears here in an essential way. The two-set principle involves one set, X, covered by two explicitly mentioned sets A and B, whereas the c-set principle involves an indexed family of c sets, which is implemented as a function which maps X into c. Actually, in our Nqthm script, the proof of Lemma 6 does not immediately follow Lemma 5, but comes after the function $\alpha \star c$ is defined (see below).

We then stuck in the definitions of "homogeneous" (`homp`) and "pre-homogeneous" (`pre-homp`). These notions are both a bit tedious to formalize on Nqthm, since they are the form $\forall S \subseteq X(\cdots)$, which must be implemented by a function which runs down a list of all subsets of X. Once the basic properties of these notions were developed, it was fairly easy to verify the *Derived Partition Lemma* (Lemma 9).

As mentioned above, we never needed to define the notion of "mesh", but we did have to prove, in what we called the *Tail Lemma*, that part of Lemma 7 which asserted that if Z is $\omega^\alpha + \omega + d$ – large, $z = \min(Z)$, and we apply `cdr` $\mu(z + d)$ times to Z, then we get an ω^α – large set $Q \subseteq Z \backslash \{z\}$ with $\min(Q) \geq \mu(z+d)$. This took about 850 lines of code to verify.

At this point, we gave the definition and some elementary properties of the functions $\varphi(\alpha, c)$ and $\alpha \star n$. We then proceeded to prove the *Cdr Lemma*, which says that the Q above is $\varphi(\hat{\alpha}, c) \cdot c^{2^z}$ – large, where $\hat{\alpha} = \{\alpha\}(z)$. This computation, which was simply embedded in the proof of Lemma 11, took about 300 lines to verify. Then came the c-set pigeon-hole principle (see above), and some elementary facts about ranges and segments.

We then proceeded to verify the Ordinal Ramsey Theorem for 1-tuples, which is the basis for the induction to come later. In §1, this was just Lemma 6 again, but there is a subtle difference. Lemma 6 involved partitioning the elements of X into c pieces, whereas now we are partitioning the 1-tuples from X. Informally, elements are often identified with 1-tuples, but it took about 300 lines of Nqthm code to handle this formally.

After some miscellaneous constructions, we then turned to the *Nice Set Lemma*, which formalizes the "equivalence relation" part of the argument in Lemma 11. In §1, we simply assumed the reader understood what it meant for the set Q to be partitioned into c^{2^x} equivalence classes, and that the pigeon-hole principle would apply to this partition, but the corresponding Nqthm proof took about 375 lines. We were then able to proceed to verify Lemma 12 (extracting a pre-homogeneous set) and then prove the Ordinal Ramsey Theorem. Finally, using ϵ_0 recursion, we defined the function Λ and proved the Paris-Harrington Ramsey Theorem.

§**3. Conclusion.** This work demonstrates that one may use Nqthm to verify some rather complex combinatorial statements. It also demonstrates one of the difficulies in doing so. Human mathematicians recognize immediately when various representations of a concepts are essentially the "same". For example, if we want to partition a set X into two pieces, a map $g : X \to 2$ is "the same as" a cover of X by disjoint sets A and B, and both these are "the same as" a partition of the set of 1-element subsets of X. We also recognize, by experience, that these identifications are only correct in certain contexts; but it is not trivial to explain formally what these contexts are. Thus, making use of these identifications is a difficult challenge for future generations of verification systems.

References

[1] Basin, D. and Kaufmann, M, The Boyer-Moore Prover and Nuprl: An Experimental Comparison Technical Report #58, Computational Logic, Inc., 1990.

[2] Boyer, R. S. and Moore, J. S., *A Computational Logic Handbook*, Academic Press, 1988.

[3] Gentzen, G., Die Widerspruchsfreiheit der reinen Zahlentheorie, *Mathematische Annalen* 112 (493 – 565) 1936.

[4] Ketonen, J. and Solovay, R., Rapidly Growing Ramsey Functions, *Annals of Math* 113 (1981) 267 – 314.

[5] Paris, J. and Harrington, L., A Mathematical Incompleteness in Peano Arithmetic, in *Handbook of Mathematical Logic*, J. Barwise, ed., North-Holland, 1978, pp. 1133 – 1142.