Some Decision Problems of Enormous Complexity

Harvey M. Friedman
Department of Mathematics
The Ohio State University
friedman@math.ohio-state.edu

Abstract

We present some new decision and comparison problems of unusually high computational complexity. Most of the problems are strictly combinatorial in nature; others involve basic logical notions. Their complexities range from iterated exponential time completeness to ε_0 *time completeness to* $\theta(\Omega^\omega,0)$ *time completeness to* $\theta(\varepsilon_{\Omega+1},0)$ *time completeness. These three ordinals are well known ordinals from proof theory, and their associated complexity classes represent new levels of computational complexity for natural decision problems. Proofs will appear in an extended version of this manuscript to be published elsewhere.*

1. Iterated exponential time - universal relational sentences

Let F be a function from A* into B*, where A,B are finite alphabets. We say that F is iterated exponential time computable if and only if there is a multitape Turing machine TM (which processes inputs from A* and outputs from B*) and an integer constant c > 0 such that TM computes F(x) with run time at most $2^{[c|x|]}$. Here $2^{[k]}$ is the exponential stack of 2's of height k and |x| is the length of the string x. More generally, $2^{[k]}(n)$ is the exponential stack of k 2's with n placed on top. Define $2^{[0]} = 1$ and $2^{[0]}(n) = n$. Hence $2^{[k]} = 2^{[k]}(1)$ and $2^{[1]}(n) = 2^n$.

The iterated exponential time computable sets strictly include those sets in the more familiar class of elementary time computable sets - where the stack of 2's is of fixed height and |x| appears at the top of the stack.

We say that X is iterated exponential time complete if and only if X is in iterated exponential time and every Y in iterated exponential time is polynomial time reducible to X. It is well known that for every finite alphabet A there exists an iterated exponential time complete X $\subseteq$ A*.

As is customary, these definitions extended to include sets of strings in a finite alphabet using characteristic functions.

A decision problem is given by a set of strings in a finite alphabet, where the "decision" is to decide membership.

Suppose we are given a map G:A* $\to \mathbb{N}$, where $\mathbb{N}$ is the set of all nonnegative integers. We can consider the associated equality problem: decide, given two strings x,y $\in$ A*, whether G(x) = G(y).

We can also consider the comparison problem: given x,y $\in$ A*, compare the numbers G(x) and G(y). We find this terminology convenient and suggestive.

Strictly speaking, this comparison problem is not a decision problem. It amounts to considering the function G'(x,y) = 0 if F(x) = F(y); 1 if F(x) < F(y); 2 if F(x) > F(y). It is easy to see that it is computationally equivalent, in the strongest possible sense, to the related decision problem: decide, given x,y $\in$ A*, whether F(x) < F(y).

A second kind of comparison problem that we consider is where we are given a set (class) valued map H on A*. Here the problem is to compare H(x) and H(y) under inclusion. Again this corresponds to a three valued function, and the problem is computationally identical to deciding the inclusion relation.

Many interesting decision problems are known to be complete in iterated exponential time. The list is substantial and includes:

term reduction in typed lambda-calculus (due to Statman)
first order theory of standard pairing functions (due to Tenney, Ferrante/Rackoff)
first order theory of a unary function (or successor function) (due to Ehrenfeucht, Rabin)
weak monadic theory of a unary function (or successor function) (due to Meyer, Rabin)
monadic theory of two successor functions (due to Rabin)
first order theory of linear orderings (due to Ehrenfeucht, Rabin, Meyer)
emptiness problem for regular expressions with complementation (due to Meyer/Stockmeyer)

Note that none of these problems are strictly combinatorial in nature. They all involve formal languages. In sections 2-6 we present a number of new combinatorial decision problems that are complete for iterated exponential time and more.

In this section we present a number of decision problems which are complete in iterated exponential time. The setting goes back to Ramsey's original paper [Ra30]. The versions in this section involve languages, but we present particularly attractive reformulations that are strictly combinatorial in terms of hypergraphs and functions in sections 2 and 3.

A universal relational sentence is a sentence of first order predicate calculus with equality which starts with one or more universal quantifiers, and is followed by a quantifier free formula with only relation symbols. Note that if M is a model of a universal relational sentence, then every submodel of M is a model of that universal relational sentence. The cardinality of a relational structure M is the cardinality of its domain. All structures are required to have a finite relational type.

If M is a model of φ then M may have a strictly larger relational type than φ. It will be convenient to define M to be a matching model of □ if and only if M satisfies □ and the relational type of M is the same as the relational type of □. In general, the number of matching models of □ (up to isomorphism) is more interesting than the number of models of □ (up to isomorphism), because the latter is always infinite, whereas the former may be finite.

Ramsey proved the following, as an application of his famous Ramsey theorem.

THEOREM 1.1. (Ramsey). Let □ be a universal relational sentence. The following are equivalent.
i) □ has models of every nonzero finite cardinality;
ii) □ has models of every nonzero cardinality;
iii) □ has a model of cardinality twice the number of distinct variables appearing in □, whose domain forms a set of atomic indiscernibles.

We need to explain iii). Let M be a model and E □ dom(M). We say that E is a set of atomic indiscernibles if and only if the following holds. There is a linear ordering < of E such that for all $n \geq 1$ and atomic formulas □$(x_1,\ldots,x_n)$ in the language of M with exactly the variables shown (repetitions allowed), and y,z in E^n that are strictly increasing under <, we have □(y) □ □(z) in M.

From Theorem 1.1, Ramsey read off a decision procedure for determining whether □ has an infinite model. In modern terms, we obtain a decision procedure that is nondeterministic exponential in the norm of □. In fact, nondeterministic exponential time completeness was established in [Le80].

The following Theorem indicates how iterated exponentiation comes up in connection with the size of models of universal relational sentences up to isomorphism.

THEOREM 1.2. (Ramsey) For $n \geq 1$, let f(n) be the least integer r such that any universal relational sentence with at most n occurrences of variables either has models of every cardinality or has all of its models of cardinality $\leq$ r. There are constants $c,d > 0$ such that for all $n \geq 1$, $2^{[[cn]]} < f(n) < 2^{[[dn]]}$.

It is easy to show that the following problem is nondeterministic exponential time computable.

Given: Two universal relational sentences, □ and □.
Decide: □ and □ have the same models.

This is because if, say, there is a model of □ that is not a model of □, then there is a model of □ that is not a model of □ whose cardinality is at most the number of distinct variables occurring in □. Using [Le80], we can see that this is nondeterministic exponential time complete.

However, consider this variant:

Given: Two universal relational sentences, □ and □.
Decide: □ and □ have the same largest models.

Here we say that M is a largest (matching) model of □ if and only if M is a (matching) model of □ and there is no (matching) model of □ with a greater number of elements. It is clear that any largest model of a universal relational sentence must be finite.

Note that this problem is decidable in iterated exponential time by Theorem 1.2. We have proved that this problem is iterated exponential time complete.

Here is a list of related decision and comparison problems that we have considered.

Given: Two universal relational sentences, □ and □.
Decide:
1. □ and □ have the same largest models.
2. □ and □ have the same number of matching models up to isomorphism.
3. □ and □ have the same cardinalities of models.
4. □ and □ have the same number of largest matching models up to isomorphism.
Compare:
5. The largest models of □ and of □, by inclusion.
6. The number of matching models of □ and of □ up to isomorphism, by magnitude.
7. The cardinalities of models of □ and of □, by inclusion.
8. The number of largest matching models of □ and of □ up to isomorphism, by magnitude.
Given: One universal relational sentence, □.
Decide:
9. □ has a unique largest matching model up to isomorphism.
10. There is a largest model of □ whose cardinality is an even integer.
11. The number of matching models of □ an even integer up to isomorphism.
12. The number of largest matching models of □ an even integer up to isomorphism.

In 3 and 7, note that by Theorem 1.1, the cardinalities of models are either a finite initial segment of the positive integers, or all nonzero cardinalities.

We have proved that 1,3-8,10 are exponential time complete. The exponential time completeness of 3 and 7 was established in [Fr84]. It is easy to see that 2,9,11,12 are iterated exponential time computable, but we don't have a significant lower bound.

Note that "does □ have a largest model?" is equivalent to □ not having an infinite model, and so is co-nondeterministic exponential time complete by [Le80].

2. Iterated exponential time - omitting hypergraphs.

Hypergraphs are studied and used in various contexts in combinatorics and computer science (e.g., in database theory). A (undirected) graph G is a pair (V,E), where V is a set of objects called vertices, and E is a set of subsets of V of cardinality 2 called edges. A hypergraph H is a pair (V,F), where V is a set of objects called vertices, and F is a set of nonempty finite subsets of V called hyperedges. The cardinality of a hypergraph is taken to be the cardinality of its set of vertices.

Let H = (V,F) be a hypergraph and $A \subseteq V$. We write H|A for the hypergraph whose vertex set is A and whose hyperedge set consists of all hyperedges of H that are subsets of A. The induced subhypergraphs of H are the hypergraphs of the form H|A, $A \subseteq V$.

There is an obvious notion of isomorphism between hypergraphs (V,F) and (V',F'). These are bijections $g:V \to V'$ such that the expression g[A] defines a bijection from F onto F'.

Let K be a finite set of finite hypergraphs. We say that a hypergraph H omits K if and only if no element of K is isomorphic to any induced subhypergraph of H.

In order to restate the twelve problems from section 1 for hypergraphs, we need the following concept. For $k \geq 1$, a k-hypergraph is a hypergraph such that every hyperedge has cardinality $\leq k$.

The restatements of the twelve problems from section 1 for hypergraphs form attractive purely combinatorial decision problems. We can establish the very same results as in section 1 for these problems. This is easily seen to amount to redoing all of the results in section 1 where all structures are required to have the following properties:

a) all predicates are symmetric;

b) all predicates hold only of distinct arguments;

c) for some $k \geq 1$, the relational type consists of exactly one predicate of each arity $\leq k$.

This restriction to structures with these three properties creates some serious difficulties which we have been able to overcome. For instance, one has a problem right from the start with the axioms for linear order, which cannot be stated or simulated (in various senses) solely in terms of universal relational sentences about such structures.

Here are the restatements.

THEOREM 2.1. Let $k \geq 1$ and K be a finite set of finite k-hypergraphs. The following are equivalent.

i) there are k-hypergraphs omitting K of every finite cardinality;

ii) there are k-hypergraphs omitting K of every cardinality;

iii) there is a uniform k-hypergraph omitting K whose number of vertices is twice the number of vertices in the largest elements of K.

We need to explain iii). Let H = (V,F) be a k-hypergraph and $E \subseteq V$. We say that H is uniform if and only if for all $1 \leq i \leq k$, either every set of i vertices is a hyperedge or no set of i vertices is a hyperedge.

From Theorem 2.1 we can read off a decision procedure for determining whether there is an infinite k-hypergraph that omits K, which runs in nondeterministic exponential time. In fact, this problem is nondeterministic exponential time complete.

THEOREM 2.2. For $k,n \geq 1$, let f(k,n) be the least integer r such that for any finite set K of finite k-hypergraphs, each with at most n vertices, either there are k-hypergraphs of every cardinality omitting K, or every k-hypergraph omitting K has at most r vertices. There are constants c,d > 0 such that for all $n \geq 1$ and $k \geq 2$, $2^{[k-2]}([cn]) < f(k,n) < 2^{[k-1]}([dn])$.

As in section 1, the following problem is nondeterministic exponential time complete, with k given in either unary or binary:

Given: An integer $k \geq 1$, and two finite sets of finite k-hypergraphs, K and S.

Decide: The k-hypergraphs omitting K and omitting S are the same.

As in section 1, consider this variant:

Given: An integer $k \geq 1$, and two finite sets of finite k-hypergraphs, K and S.

Decide: The largest k-hypergraphs omitting K and omitting S are the same.

Here we say that H is a largest k-hypergraph omitting K if and only if H is a k-hypergraph omitting K and no k-hypergraph of larger cardinality omits K. It is clear that any largest k-hypergraph omitting K must be finite.

We have proved that this problem is iterated exponential time complete.

Here is a list of related problems we have considered:

Given: An integer $k \geq 1$, and two finite sets of finite k-hypergraphs, K and S.

Decide:

1. The largest k-hypergraphs omitting K and omitting S are the same.
2. The number of k-hypergraphs omitting K and omitting S up to isomorphism are the same.
3. The cardinalities of k-hypergraphs omitting K and omitting S are the same.
4. The cardinalities of largest k-hypergraphs omitting K and omitting S are the same.

Compare:

5. The largest k-hypergraphs omitting K and omitting S, by inclusion.
6. The number of k-hypergraphs omitting K and omitting S up to isomorphism, by magnitude.
7. The cardinalities of k-hypergraphs omitting K and omitting S, by inclusion.
8. The number of largest k-hypergraphs omitting K and omitting S up to isomorphism, by magnitude.

Given: An integer $k \geq 1$ and a finite set of finite k-hypergraphs, K.

Decide:

9. There is a unique largest k-hypergraph omitting K up to isomorphism.
10. There is a largest k-hypergraph omitting K whose cardinality is an even integer.
11. The number of k-hypergraphs omitting K is an even integer.
12. The number of largest k-hypergraphs omitting K is an even integer up to isomorphism.

In 3 and 7, note that by Theorem 1.1, the cardinalities of k-hypergraphs omitting K are either a finite initial segment of the positive integers, or all cardinalities.

We have proved that 1,3-8,10 are exponential time complete, where k is presented either in unary or binary. It is easy to see that 2,9,11,12 are iterated exponential time computable, but we don't have a significant lower bound, whether k is in unary or binary.

Note that "there is a largest k-hypergraph omitting K" is equivalent to "there is no infinite k-hypergraph omitting K," and so is in co-nondeterministic exponential time by Theorem 2.1, with k in unary or binary. In fact, it is co-nondeterministic exponential time complete.

3. Iterated exponential time - omitting functions.

We present results for functions that correspond to the results for hypergraphs in section 2. Moving to functions is a natural step to take in preparation of section 5, where we take a big jump in computational complexity.

A k-ary function is a function whose domain is of the form A^k, where A is any set. If the domain is A^k then we define the 1-domain to be A. The size of a k-ary function is defined to be the cardinality of its 1-domain. A k-ary function is finite if and only if its domain is finite if and only if its 1-domain is finite. We write dom(f) for the domain of f and 1-dom(f) for the 1-domain of f.

Let f be a k-ary function and E $\subseteq$ 1-dom(f). We write f|E for the restriction of f to E^k. Thus f|E $\subseteq$ f, dom(f|E) = E^k, and 1-dom(f|E) = E.

Let f,g be k-ary functions. We say that f,g are isomorphic if and only if there is a bijection h from 1-dom(f) onto 1-dom(g) such that for all $x_1,\ldots,x_{2n} \in$ 1-dom(f), $f(x_1,\ldots,x_n) = f(x_{n+1},\ldots,x_{2n})$ if and only if $g(h(x_1),\ldots,h(x_n)) = g(h(x_{n+1}),\ldots,h(x_{2n}))$. This notion of isomorphism, which is one of many, is particularly appropriate for this study.

Let K be a set of k-ary functions. We say that a k-ary function f omits K if and only no element of K is isomorphic to any restriction of f.

THEOREM 3.1. Let k ≥ 1 and K be a finite set of finite k-ary functions. The following are equivalent.

i) there are k-ary functions omitting K of every finite size;

ii) there are k-ary functions omitting K of every size;

iii) there is a k-ary function omitting K whose size is four times the size of the largest elements of K, and whose 1-domain forms a set of atomic indiscernibles.

We need to explain iii). Let f be a k-ary function and E $\subseteq$ 1-dom(f). We say that E is a set of atomic indiscernibles if and only if the following holds. There is a linear ordering < of E such that for all $x_1,\ldots,x_{2n},y_1,\ldots,y_{2n} \in$ E, if $(x_1,\ldots,x_{2n})$ and $(y_1,\ldots,y_{2n})$ have the same order type under <, then $f(x_1,\ldots,x_n) = f(x_{n+1},\ldots,x_{2n})$ if and only if $f(y_1,\ldots,y_n) = f(y_{n+1},\ldots,y_{2n})$.

From Theorem 3.1 we can read off a decision procedure for determining whether there is an infinite function that omits K, which runs in nondeterministic exponential time. In fact, it is nondeterministic exponential time complete.

THEOREM 3.2. For k,n ≥ 1, let f(k,n) be the least integer r such that for any finite set K of finite k-ary functions, each of size at most n,, either there are k-ary functions of every cardinality omitting K, or every k-ary function omitting K has size at most r. There are constants c,d > 0 such that for all n ≥ 1 and k ≥ 2, $2^{[k-2]}([cn]) < f(k,n) < 2^{[2k-1]}([dn])$.

We have not attempted to tighten up the above bounds.

The following problem is nondeterministic exponential time complete, with k given in either unary or binary, using [Le80].

Given: An integer k ≥ 1, and two finite sets of finite k-ary functions, K and S.

Decide: The k-ary functions omitting K and omitting S are the same.

Now consider this variant:

Given: An integer k ≥ 1, and two finite sets of finite k-ary functions, K and S.

Decide: The largest k-ary functions omitting K and omitting S are the same.

Here we say that f is a largest k-ary function omitting K if and only if f is a k-ary function omitting K and no k-ary function of larger size omits K. It is clear that any largest k-ary function omitting K must be finite.

We have proved that this problem is iterated exponential time complete.

Here is a list of related problems we have considered.

Given: An integer k ≥ 1, and two finite sets of finite k-ary functions, K and S.

Decide:

1. The largest k-ary functions omitting K and omitting S are the same.
2. The number of k-ary functions omitting K and omitting S up to isomorphism are the same.
3. The sizes of k-ary functions omitting K and omitting S are the same.
4. The sizes of largest k-ary functions omitting K and omitting S are the same.

Compare:

5. The largest k-ary functions omitting K and omitting S, by inclusion.
6. The number of k-ary functions omitting K and omitting S up to isomorphism, by magnitude.
7. The sizes of k-ary functions omitting K and omitting S, by inclusion.
8. The number of largest k-ary functions omitting K and omitting S up to isomorphism, by magnitude.

Given: An integer $k \geq 1$ and a finite set of finite k-ary functions, K.

Decide:

9. There is a unique largest k-ary function omitting K up to isomorphism.
10. There is a largest k-ary function omitting K whose size is an even integer.
11. The number of k-ary functions omitting K is an even integer.
12. The number of largest k-ary functions omitting K is an even integer up to isomorphism.

The versions with k given in unary and k given in binary are evidently computationally equivalent.

In 3 and 7, note that by Theorem 3.1, the sizes of k-ary functions omitting K are either a finite initial segment of the positive integers, or all cardinalities.

We have proved that 1,3-8,10 are exponential time complete. It is easy to see that 2,9,11,12 are iterated exponential time computable, but we don't have a significant lower bound.

The problem "there is a largest k-ary function omitting K" is equivalent to the problem "there is no infinite k-ary function omitting K," and is co-nondeterministic exponential time complete using [Le80]. Here k can be given in unary or binary.

4. Ordinal recursion - a robust approach.

Let <' be a strict well ordering of a subset A of $\mathbb{N}$ of order type α. For each k, we define $(n_1,\ldots,n_k)$ $<'_{lex}$ $(m_1,\ldots,m_k)$ if and only if $n_i <' m_i$ for the least i such that $n_i \neq m_i$.

We now define the nested <' multirecursive functions. They form the least family S of functions of several variables on $\mathbb{N}$ into $\mathbb{N}$ such that the following holds:

Let $k \geq 1$ and t be a term involving the k-ary function symbol F', variables $n_1,\ldots,n_k$, constants for elements of $\mathbb{N}$, elementary functions, and functions in S. Let t' be such a term not involving F'. Then the unique function $F:\mathbb{N}^k \to \mathbb{N}$ defined by the equation

$$F(n_1,\ldots,n_k) = t(F',n_1,\ldots,n_k)$$

lies in S, where each subterm $F'(s_1,\ldots,s_k)$ of t is interpreted as:

$F(s_1,\ldots,s_k)$ if $(s_1,\ldots,s_k)$ $<'_{lex}$ $(n_1,\ldots,n_k)$; $t'(s_1,\ldots,s_k)$ otherwise.

It will be useful to define the truncated nested <' multirecursive functions. They form the least family S of functions of several variables from $\mathbb{N}$ into $\mathbb{N}$ such that the following holds:

Let $k \geq 1$ and $p \geq 0$ and $b \in A$ and t be a term involving the k-ary function symbol F', variables $n_1,\ldots,n_{k+p}$, constants for elements of $\mathbb{N}$, elementary functions, and functions in S. Let t' be such a term not involving F'. Then the unique function $F:\mathbb{N}^k \to \mathbb{N}$ defined by the equation

$$F(n_1,\ldots,n_{k+p}) = t(F',n_1,\ldots,n_{k+p})$$

lies in S, where each subterm $F'(s_1,\ldots,s_{k+p})$ of t is interpreted as: $F(s_1,\ldots,s_{k+p})$ if $(s_1,\ldots,s_k)$ $<'_{lex}$ $(n_1,\ldots,n_k)$ and $s_1,\ldots,s_k,n_1,\ldots,n_k <' b$; $t'(s_1,\ldots,s_{k+p})$ otherwise.

(Here the elementary functions from $\mathbb{N}^k \to \mathbb{N}$ are the functions that can be computed in time at most $2^{[n]}(\max(x))$, where n is some fixed integer, and x is the input.)

Note that if <" is another strict well ordering of type α and there is an elementary bijection of $\mathbb{N}$ which maps <' onto <" and has an elementary inverse, then the nested <' multirecursive functions are the same as the nested <" multirecursive functions, and the truncated nested <' multirecursive functions are the same as the truncated nested <" multirecursive functions.

We now relate our definitions for a standard choice of <' of type ϵ_0 to usual approaches in the literature. We refer the reader to [Ro84] and its references.

We use the standard Cantor normal form for ordinals < ϵ_0. We can think of these as strings in a finite alphabet. Using any elementary recursive surjective pairing function on $\mathbb{N}$ with elementary inverse, we can map these strings onto $\mathbb{N}$, thereby getting a well ordering of type ϵ_0, but using only some elements of $\mathbb{N}$. It is easy to see that there is an elementary bijection from the elements of $\mathbb{N}$ used and all of $\mathbb{N}$, whose inverse is elementary. Using this bijection, we obtain our $<'(\epsilon_0)$ of type ϵ_0. It is also easy to see that all <' obtained in this way are isomorphic by an elementary bijection with elementary inverse. By the preliminary remark above, the (truncated) nested <'-multirecursive functions are the same for all such <'. We refer to (any such) <' as $<'(\epsilon_0)$.

THEOREM 4.1. The following classes of functions are equal.

i) the truncated nested $<'(\epsilon_0)$ multirecursive functions as defined above;

ii) the functions in the Grzegorczyk hierarchy up to ϵ_0, as defined on page 80 of [Ro84], using standard fundamental sequences for ϵ_0;

iii) the functions in the Wainer hierarchy up to ϵ_0, as defined on page 84 of [Ro84], using standard fundamental sequences for ϵ_0;

iv) the $<\epsilon_0$ recursive functions as defined on page 89 of [Ro84], using $<'(\epsilon_0)$;

v) the provably recursive functions of Peano Arithmetic.

The standard fundamental sequences for ϵ_0 are defined on page 78 of [Ro84]. In the Grzegorczyk hierarchy, each successive stage amounts to a single primitive recursion. Thus the union of the first ω levels is exactly the primitive recursive functions. In the Wainer hierarchy, each stage amounts to taking the Hardy function at that stage and closing under limited recursion and limited composition. In ordinal recursion (iv), the recursion is only on a single variable and is unnested. Functions obtained in this way are used in further recursions.

Let <'(□) be the usual ordering on □. The nested <'(□) multirecursive functions are already very extensive; e.g., far beyond the primitive recursive functions, as we see from the following.

THEOREM 4.2. The following classes of functions are equal.
i) the nested <'(□) multirecursive functions as defined above;
ii) the functions in the Grzegorczyk hierarchy up to $□^{□}$, using standard fundamental sequences for $□^{□}$;
iii) the functions in the Wainer hierarchy up to $□^{□^{□}}$, using standard fundamental sequences for $□^{□^{□}}$;
iv) the $<□^{□^{□}}$ recursive functions, using $<'(□_0)$, or a truncated version for $□^{□^{□}}$;
v) the functions that are k-recursive for some k ≥ 1, as defined on page 16 of [Ro84];
vi) the provably recursive functions of 2 quantifier induction.

For the sake of completeness, we present the following well known result which places the primitive recursive functions in context:

THEOREM 4.3. The following classes of functions are equal.
i) the primitive recursive functions;
ii) the functions in the Grzegorczyk hierarchy up to □;
iii) the functions in the Wainer hierarchy up to $□^{□}$, using standard fundamental sequences for $□^{□}$;
iv) the provably recursive functions of 1 quantifier induction, or RCA_0 (see [Si99], p. 23).

We now wish to define the <' time computable functions. For this purpose, we define the concept of a nested <' multirecursive derivation. This is a finite list of applications of the clause we used that defines the nested <' multirecurive functions, where every function used in each application (other than the function introduced) is either explicitly elementary or is a function that is previously introduced. When we say "explicitly elementary" we mean that a Turing machine code and a height of the exponential stack is considered part of the list.

In order to define the <' time computable functions, we actually use the related concept of a truncated nested <' multirecursive derivation. This is a finite list of applications of the clause we used that defines the truncated nested <' multirecursive functions, where every function used in each application (other than the function introduced) is either explicitly elementary or is a function that is previously introduced, and where we explicitly designate the b's that are used.

Note that every truncated nested <' multirecursive function amounts to a finite string from the elements of □ (plus a finite number of auxiliary letters). So we can define its size as the max of all elements of □ appearing plus the total length. We then define NMR(<'):□ □ □ by NMR(<')(n) = the maximum value of any truncated nested <' multirecursive function at any arguments ≤ n given by a truncated nested <' multirecursive derivation of size ≤ n. (Here NMR stands for "nested multirecursion").

Finally, we say that a function on finite strings in a finite alphabet is in <' time if and only if there is a constant c such that it can be computed in time complexity NMF(<')(cn). A function F on finite strings in a finite alphabet is said to be <' time complete if and only if every function in <' time is polynomial time reducible to F. It is easy to see that <' time and <' time completeness does not change under an elementary isomorphism with elementary inverse.

We write $□_0$ time and $□_0$ time complete to indicate $<'(□_0)$ time and $<'(□_0)$ time complete.

For the higher so called "proof theoretic ordinals" □, there is an associated "notation system" whose order type is □. For our purposes, this amounts to an elementary set E of finite strings in a finite alphabet together with an elementary ≤ relation on that set E, where ≤ defines a pre well ordering; i.e., a well ordering except that we may have x ≤ y and y ≤ x for distinct x and y.

Obviously, there is an elementary map h from □ into E such that x <' y if and only if h(x) ≤ h(y) and h(x) ≠ h(y). We have thus turns such a pre well ordering into a strict well ordering <' on □.

It is easy to see that we have the following robustness. Any two <' obtained in this way give rise to the same nested <' multirecursive functions, the same truncated nested <' multirecursive functions, the same <' time computable functions, and the same <' time complete functions.

Thus, if we accept a given presentation of a proof theoretic ordinal □, then we have already robustly determined the nested □ multirecursive functions, the truncated nested □ multirecursive functions, the □ time computable functions, and the □ time complete functions.

But what is so special about the current presentations of proof theoretic ordinals □? Or what is so special about some elementary well orderings of type □? We make some progress on this important question for certain ordinals (ordinals less than $□(□^{□},0)$ in the Feferman Aczel notations; see references below). We feel that this question has a very good answer up through all of the proof theoretic ordinals that have so called natural notation systems in the current literature. But many deep issues remain even below $□(□^{□},0)$. We now discuss our approach to this matter in some detail.

Certain countable ordinals are of great interest to combinatorists, proof theorists, and theoretical computer scientists. These generally go under the name of "proof theoretic ordinals" because of their original connections with key formal systems. But they have become important in rewriting systems, wqo theory, and elsewhere.

For each of these ordinals, □, there is an associated collection of finitely many constants and functions on □ such that every ordinal < □ is given by a term. This provides what we call a representation of the ordinal. (This may also be called a notation system, but this usually refers to a system with some additional structure such as the identification of normal forms.) For example, for □, there is the representation (□,0,□+1). For $□^{□}$, there is the representation $(\square^{\square},0,1,\square,x+y,x\bullet y)$. For $\square_0$, there is the representation $(\square_0,0,\square^{\square}+\square)$. For $\square_0$, there is the representation $(\square_0,0,\square+\square,\square(\square,\square))$, where $\square(0,\square) = \square^{\square}$, and for each $\square$, $\square(\square+1,\square)$ enumerates the fixed points of $\square(\square,\square)$ as a function of $\square$, and for limits $\square$, $\square(\square,\square)$ is the sup of the $\square(\square,\square)$, $\square < \square$.

Sometimes an infinite list of constants indexed by $\square$ is used in notation systems; e.g., constants $\square$, $\square_1$, $\square_2$, etc. But this naturally lends itself to an essentially equivalent representation that is in a finite relational type; here by introducing the cardinal successor function and using only $\square$. Also such ordinals can be viewed as proper initial segments of larger ordinals for which the usual representations are in a finite relational type.

Note that all of the functions used in the above examples of representations are dominating and increasing. I.e., $f(\square_1,\ldots,\square_n) \geq \square_1,\ldots,\square_n$, and if $\square_1 \leq \square_1,\ldots,\square_n \leq \square_n$, then $f(\square_1,\ldots,\square_n) \leq f(\square_1,\ldots,\square_n)$. The ordinals that can be generated by a system of finitely many such ordinal functions has been calculated in [Schm75]. The calculations there are in terms of Schutte's Klammersymbols. In terms of the more customarily used Feferman Aczel notations, they are the ordinals $< \square(\square^{\square},0)$. The Feferman Aczel notations are discussed in, e.g., [Schu77], p. 224, and [Bu86].

An algebraic type $\square$ consists of a description of how many constant symbols, how many 1-ary function symbols, how many 2-ary function symbols, etcetera, where there are only finitely many function symbols, and there is at least one constant symbol.

In an unpublished 1984 manuscript, we proved that there is a largest ordinal $o(\square)$ that has a dominating and increasing representation in any given algebraic type $\square$. And we can compute $o(\square)$ very simply in the Feferman Aczel notations. The fragment of the Feferman Aczel notations for $\square(\square^{\square},0)$ can be described in particularly simple terms involving functions on countable ordinals only.

$o(\square)$ is also the largest order type of any dominating and increasing structure $(\square,<',\ldots)$ of type $\square$, where dominating and increasing refers to the linear ordering <' on $\square$, and every element of $\square$ is given by a closed term. This follows from the previous paragraph by the fundamental result of [Hi52] to the effect that in all such structures, <' is well ordered.

We now introduce the elementary generating systems. An elementary generating system of algebraic type $\square$ consists of a structure $M = (\square,<')$ such that

i) M is a structure of type $\square$ with domain $\square$;
ii) <' is an elementary linear ordering on $\square$;
iii) every function of M is elementary;
iv) there is a k such that for every $n \in \square$ there is a closed term in M with at most $2^{[k]}(n)$ occurrences of function symbols whose value is n.

We say that an elementary generating system is dominating and increasing if its functions are dominating and increasing with respect to <'.

THEOREM 4.4. Let $\square$ be an algebraic type which contains at least one function symbol of arity ≥ 2. Among the dominating and increasing elementary generating systems of type $\square$ there is one whose set of truncated nested <' multirecursive functions is maximum (where <' is its linear ordering). All such systems must be of order type $o(\square)$. Also, there is a dominating and increasing elementary generating system of type $\square$ such that every dominating and increasing elementary generating system of type $\square$ can be embedded into it by an elementary order preserving map. Any such system must be maximum in the first sense. In addition, the usual notation systems from proof theory provide examples which is maximum in both senses. The same results hold if we replace "truncated nested <' multirecursive functions" by "<' time computable functions."

We conjecture that in any algebraic type there is a dominating and increasing elementary generating system of algebraic type $\square$, order type $o(\square)$, and not maximum in either of the two senses (or three senses if we include <' time).

Theorem 4.4 gives an appropriate meaning to truncated nested $\square$ multirecursive and $\square$ time and $\square$ time complete for ordinals $\square$ of the form $o(\square)$, where $\square$ is an algebraic type which contains at least one function symbol of arity ≥ 2. This requirement on $\square$ guarantees that $o(\square)$ is an epsilon number.

Among the ordinals $o(\square)$, where $\square$ has at least one function symbol of arity ≥ 2, one finds such familiar ordinals as the first $\square$ epsilon numbers, the first $\square$ gamma numbers, as well as every $\square(\square,n)$, where $\square < \square^{\square}$ has Cantor normal form to base $\square$ whose coefficients are all finite, and $n < \square$.

It also appears that the standard representations of the $o(\square)$ have even stronger maximumness properties involving initial segments.

As indicated above, this theory stops at $\square(\square^{\square},0)$, which is exactly one of the ordinal we need for section 6. Thus we fall back on a standard notation system for $\square(\square^{\square},0)$, such as Feferman Aczel. We pass to any appropriate <' for the notation system, and thus we have our working definitions of the (truncated) $\square(\square^{\square},0)$ multirecursive functions, $\square(\square^{\square},0)$ time computable functions, and $\square(\square^{\square},0)$ complete functions.

The restricted $\square(\square^{\square},0)$ multirecursive functions are known to be the provably recursive functions of $\square^1_2$-TI_0, which is ACA_0 together with the scheme of transfinite induction with respect to $\square^1_2$ formulas. Also, $\square(\square^{\square},0)$ is the provable ordinal of $\square^1_2$-TI_0. For a proof, see [RW93], where this system is called $\square^1_2$-BI_0, where BI stands for “bar induction.”

The much higher proof theoretic ordinal $\square(\square_{\square},0)$ is the provable ordinal of $\square^1_1$-CA_0. And the restricted $\square(\square_{\square},0)$ multirecursive functions are the provably recursive functions of $\square^1_1$-CA_0. See [BFPS81], p. 334, or [Schu77], Chapter IX.

5. $\square_0$ time - omitting functions in N.

We bring in the ordering of N = the set of all nonnegative integers, which causes an enormous jump in computational complexity.

Let $k \geq 1$. A k-ary function in N is a function of the form $f:[0,n)^k$ □ N, where n □ N.

Let f,g be k-ary functions in N. We say that f,g are order isomorphic if and only there is a one-one order preserving map h:1-dom(f) □ 1-dom(g) such that for all $x_1,\ldots,x_{2k}$ □ 1-dom(f), $f(x_1,\ldots,x_k) < f(x_{k+1},\ldots,x_{2k})$ if and only if $f(h(x_1),\ldots,h(x_k)) < f(h(x_{k+1},\ldots,x_{2k}))$.

Let K be a set of k-ary functions in N. We say that f order omits K if and only if no element of K is order isomorphic to a restriction of f.

THEOREM 5.1. Let $k \geq 1$ and K be a finite set of finite k-ary functions in N. The following are equivalent.

i) there is an infinite k-ary function in N order omitting K;

ii) there is a k-ary function order omitting K whose size is four times the size of the largest element of K, and whose 1-domain forms a set of special indiscernibles.

Here, we will not go into an explanation of the special indiscernibles needed for ii). We will just say that we can read off the nondeterministic exponential time completeness of "there is an infinite k-ary function in N order omitting K."

Note that we did not include "there are k-ary functions in N order omitting K of every finite size" in Theorem 5.1. This is not equivalent.

Theorem 5.1 is metamathematically quite significant. We can show that there is no algorithm such that ACA_0 proves is a decision procedure for "there is an infinite k-ary function in N order omitting K."

We can consider the same comparison and decision problems that we considered in section 3, where we are given one or two finite sets of finite k-ary functions in N, and where "largest k-ary function(s)" is replaced by "largest finite k-ary function(s) in N." We obtain the same results. Thus we are still in the realm of iterated exponential time.

We now introduce a crucial restriction. We say that f is a limited k-ary function if and only if f is a k-ary function in N such that for all x □ dom(f), $f(x) \leq \max(x)$. Another restriction that leads to the same complexity findings is subexponentiality: f is a subexponential k-ary function if and only if f is a k-ary function in N such that for all x □ dom(f), $f(x) \leq 2^{\max(x)}$.

THEOREM 5.2. Let $k \geq 1$ and K be a finite set of finite k-ary functions in N. The following are equivalent.

i) there are limited k-ary functions omitting K of every finite size;

ii) there is an infinite limited k-ary function omitting K;

iii) there is a limited k-ary function omitting K whose size is four times the size of the largest elements of K, and whose 1-domain forms a set of very special indiscernibles.

Again, we won't describe the very special indiscernibles needed for iii). We will just say that we can read off the nondeterministic exponential time completeness of "there is an infinite limited k-ary function order omitting K."

THEOREM 5.3. Let f(k,n) be the least integer r such that for any finite set K of k-ary functions in N, each of size $\leq n$, either there is an infinite limited k-ary function which order omits K, or every limited k-ary function which order omits K has size $\leq r$. Then f is □$_0$ time complete. Also the function f(k,k) eventually dominates every truncated □$_0$ multirecursive function.

Consider the following decision problem:

Given: An integer $k \geq 1$, and two finite sets K,S, of finite k-ary functions in N.
Decide: The finite limited k-ary functions omitting K and omitting S are the same.

As opposed to the analogous earlier situations, we do not understand this problem. We don't even know whether it is decidable. We conjecture that it is nondeterministic exponential time complete.

Consider the following variant:

Given: An integer $k \geq 1$, and two finite sets K,S, of finite k-ary functions in N.
Decide: The largest finite limited k-ary functions omitting K and omitting S are the same.

We have proved that this problem is □$_0$ time complete.

Here is a list of related problems we have considered.

Given: An integer $k \geq 1$, and two finite sets K,S, of finite k-ary functions in N.
Decide:
1. The largest finite limited k-ary functions omitting K and omitting S are the same.
2. The number of finite limited k-ary functions omitting K and omitting S up to isomorphism are the same.
3. The sizes of finite limited k-ary functions omitting K and omitting S are the same.
4. The sizes of largest finite limited k-ary functions omitting K and omitting S are the same.
5. The finite limited k-ary functions omitting K and omitting S are the same.
Compare:
6. The largest finite limited k-ary functions omitting K and omitting S, by inclusion.
7. The number of finite limited k-ary functions omitting K and omitting S up to isomorphism, by magnitude.
8. The sizes of finite limited k-ary functions omitting K and omitting S, by inclusion.
9. The number of largest finite limited k-ary functions omitting K and omitting S up to isomorphism, by magnitude.
10. The finite limited k-ary functions omitting K and omitting S.

Given: An integer $k \geq 1$ and a finite set K of finite k-ary functions in N.

Decide:
11. There is a unique largest finite limited k-ary function omitting K up to isomorphism.
12. There is a largest finite limited k-ary function omitting K whose size is an even integer.
13. The number of finite limited k-ary functions omitting K is an even integer.
14. The number of finite limited largest k-ary functions omitting K is an even integer up to isomorphism.

The versions with k given in unary and k given in binary are evidently computationally equivalent.

We have proved that 1,3,4,6-9,12 are ε_0 time complete. It is easy to see that 2,11,13,14 are ε_0 time computable, but we don't have a significant lower bound. And we don't even know whether 5 and 10 are decidable, although we strongly suspect that they are nondeterministic exponential time complete.

The problem "there is a largest limited k-ary function omitting K" is equivalent to the problem "there is no infinite limited k-ary function omitting K," and is co-nondeterministic exponential time complete using [Le80]. Here k can be given in unary or binary.

6. $\theta(\Omega^{\omega},0)$ and $\theta(\Omega_{\omega},0)$ time - tight trees.

We will consider finite trees, which are finite partial orderings with a minimum element (root). The valence of a vertex in a tree T is the number of its immediate successors. The valence of T is the largest of the valences of its vertices. The set of vertices of T is denoted by V(T).

Note that for any finite tree T = (V(T),≤), and for any vertices x,y in V(T), x inf y exists.

Let T_1 and T_2 be finite trees. An inf preserving embedding is a one-one map $h:V(T_1) \rightarrow V(T_2)$ such that for all $x,y \in V(T_1)$, h(x inf y) = h(x) inf h(y).

Recall the classic theorem of J.B. Kruskal [Kr60]:

THEOREM 6.1. Let $T_1,T_2,\ldots$ be an infinite sequence of finite trees. There exists i < j such that T_i is inf preserving embeddable into T_j.

The simplest proof is in [NW63].

The i-th truncation of a finite tree T is the subtree of T consisting of all vertices with at most i strict predecessors. Here i ≥ 0, and the 0-th trunction consists of just the root. A nonzero truncation is a truncation other than the 0-th truncation.

The peaks of a tree are those vertices such that no other vertex has more strict predecessors. A function from vertices into vertices is said to lift a vertex if that vertex is sent to a vertex with a greater number of strict predecessors.

A finite tree T is said to be tight if and only if there is no inf preserving embedding from any nonzero truncation of T into T that lifts every peak of the truncation.

THEOREM 6.2. For all k ≥ 1 there are only finitely many tight trees of valence ≤ k.

[Fr99], section 3 contains a proof of Theorem 6.2 as well as a proof of the following:

THEOREM 6.3. As a function of k ≥ 1, the number of tight trees of valence ≤ k eventually dominates every restricted nested $\theta(\Omega^{\omega},0)$ multirecursive function. As a function of k ≥ 1, the number of tight trees of valence ≤ k is $\theta(\Omega^{\omega},0)$ time computable, and in fact $\theta(\Omega^{\omega},0)$ time complete.

Note that the number of tight trees of valence ≤ k is a very simple definition of extremely large finite sets of finite objects (indexed by an integer k). They are essentially the first such examples. But what about decision problems necessarily involving extremely large computer time?

In (full) second order logic, one uses variables ranging over the domain, but also variables ranging over relations on the domain of any given arity. Thus there are variables ranging over 1-ary relations on the domain, over 2-ary relations on the domain, etcetera.

Second order properties of trees are given by formulas in second order logic using the tree partial ordering as the only nonlogical symbol.

Given: A second order property P of trees and an integer k ≥ 1.

Decide:

Does P hold of some tight tree of valence ≤ k?

THEOREM 6.4. This decision problem is $\theta(\Omega^{\omega},0)$ time complete. This is true if k is given in unary or in binary.

We can also use trees with finitely many labels and "inf preserving embeddings with gap condition," as in [Fr99], section 5. See [Si85] and [FRS87] for a discussion of our gap condition.

Let k,n ≥ 1. A gap tight tree with n labels is a tree T with a labeling function into {1,...,n}, such that there is no inf preserving embedding with the gap condition from a nonzero truncation of T into T which lifts every peak of the truncation.

We obtain the exact analogs of Theorems 6.2 and 6.3, only with the much larger proof theoretic ordinal $\theta(\Omega_{\omega},0)$, which is the ordinal of Π^1_1-CA_0, or ID(<ω); e.g., see [Schu77], Chapter IX, or [BFPS81], p. 334.

Given: A second order property P of trees and integers k,n ≥ 1.

Decide:

Does P hold of some gap tight tree of valence ≤ k with n labels.

THEOREM 6.5. This problem is $\theta(\Omega_{\omega},0)$ time complete, where k,n are given in unary or binary. It remains $\theta(\Omega_{\omega},0)$ complete, even if k is fixed to be 2. Furthermore, there exist constants c,d > 0 such that for fixed n, this problem is $\theta(\Omega^{[cn]},0)$ time hard and $\theta(\Omega^{[dn]},0)$ time easy. Here k may be given in binary, or may be fixed to be 2.

REFERENCES

[BFPS81] W. Buchholz, S. Feferman, W. Pohlers, W. Sieg: *Iterated inductive definitions and subsystems of analysis*, Lecture Notes in Math. 897, Springer Verlag, Berlin 1981.

[Bu86] W. Buchholz, "A new system of proof theoretic ordinal functions," *Annals of Pure and Applied Logic* 32, 195-207.

[Fr84] H. Friedman, "On the spectra of universal relational sentences," *Information and Control*, vol. 62, nos. 2/3, 1984, 205-209.

[FRS87] H. Friedman, N. Robertson, P. Seymour, "The metamathematics of the graph minor theorem," S. Simpson (ed.), *Logic and Combinatorics*, Contemporary Mathematics 65, American Mathematical Society, 1987, 229-261.

[Fr99] H. Friedman, 27:FiniteTrees/ predicativity: Sketches, January 13, 1999, FOM e-mail list, archived at http://www.math.psu.edu/simpson/fom/, and http://www.math.ohio-state.edu/ foundations/manuscripts.html.

[Hi52] G. Higman, "Ordering by divisibility in abstract algebras," *Proc. Lond. Math. Soc*. 2 (1952), 326-336.

[Kr60] J.B. Kruskal, "Well-quasi-ordering, the tree theorem, and Vazsonyi's conjecture," *Trans. Amer. Math. Soc.* 95 (1960), 210-225.

[Le80] H. Lewis, "Complexity results for classes of quantificational formulas," *J. of Computer and System Sciences*, 21 (1980), 317-353.

[NW63] C.St.J.A. Nash-Williams, "On well-quasi-ordering finite trees," *Proc. Cambridge Phil. Soc.* 59 (1963), 833-835.

[Ra30] F.P Ramsey, "On a problem of formal logic," *Proc. London Math. Soc.* 30, 264-286.

[RW93] M. Rathjen and A. Weiermann, "Proof-theoretic investigations on Kurskal's theorem," *Annals of Pure and Applied Logic* 60 (1993), 49-88.

[Ro84] H.E. Rose, *Subrecursion: Functions and hierarchies*, Oxford Logic Guides 9, Clarendon Press, Oxford, 1985.

[Schm75] Diana Schmidt, "Bounds for the closure ordinals of replete monotonic increasing functions," *Journal of Symbolic Logic*, volume 40, Number 3, 1975, 305-316.

[Schu77] K. Schutte, *Proof Theory*, Springer Verlag, 1977.

[Si85] S. Simpson, "Nonprovability of certain combinatorial properties of finite trees," in: *Harvey Friedman's Research on the Foundations of Mathematics,* Studies in Logic and the Foundations of Mathematics, volume 117, North-Holland, 1985, 87-117.

[Si99] S. Simpson, *Subsystems of second order arithmetic*, Perspectives in Mathematical Logic, Springer Verlag, 1999.