

Kolmogorov Complexity and the Recursion Theorem

Bjørn Kjos-Hanssen¹, Wolfgang Merkle² and Frank Stephan^{*3}

¹ University of Connecticut, Storrs, bjorn@math.uconn.edu

² Ruprecht-Karls-Universität Heidelberg, merkle@math.uni-heidelberg.de

³ National University of Singapore, fstephan@comp.nus.edu.sg

Abstract. We introduce the concepts of complex and autocomplex sets, where a set A is complex if there is a recursive, nondecreasing and unbounded lower bound on the Kolmogorov complexity of the prefixes (of the characteristic sequence) of A , and autocomplex is defined likewise with recursive replaced by A -recursive. We observe that exactly the autocomplex sets allow to compute words of given Kolmogorov complexity and demonstrate that a set computes a diagonally nonrecursive (DNR) function if and only if the set is autocomplex. The class of sets that compute DNR functions is intensively studied in recursion theory and is known to coincide with the class of sets that compute fixed-point free functions. Consequently, the Recursion Theorem fails relative to a set if and only if the set is autocomplex, that is, we have a characterization of a fundamental concept of theoretical computer science in terms of Kolmogorov complexity. Moreover, we obtain that recursively enumerable sets are autocomplex if and only if they are complete, which yields an alternate proof of the well-known completeness criterion for recursively enumerable sets in terms of computing DNR functions.

All results on autocomplex sets mentioned in the last paragraph extend to complex sets if the oracle computations are restricted to truth-table or weak truth-table computations, for example, a set is complex if and only if it wtt-computes a DNR function. Moreover, we obtain a set that is complex but does not compute a Martin-Löf random set, which gives a partial answer to the open problem whether all sets of positive constructive Hausdorff dimension compute Martin-Löf random sets.

Furthermore, the following questions are addressed: Given n , how difficult is it to find a word of length n that (a) has at least prefix-free Kolmogorov complexity n , (b) has at least plain Kolmogorov complexity n or (c) has the maximum possible prefix-free Kolmogorov complexity among all words of length n . All these questions are investigated with respect to the oracles needed to carry out this task and it is shown that (a) is easier than (b) and (b) is easier than (c). In particular, we argue that for plain Kolmogorov complexity exactly the PA-complete sets compute incompressible words, while the class of sets that compute words of maximum complexity depends on the choice of the universal Turing machine, whereas for prefix-free Kolmogorov complexity exactly the complete sets allow to compute words of maximum complexity.

* F. Stephan is supported in part by NUS grant number R252-000-212-112.

1 Introduction and overview

The Recursion Theorem, one of the most fundamental results of theoretical computer science, asserts that — with a standard effective enumeration $\varphi_0, \varphi_1, \dots$ of all partial recursive functions understood — every recursive function g has a fixed point in the sense that for some index e , the partial recursive functions φ_e and $\varphi_{g(e)}$ are the same. The question of which type of additional information is required in order to compute a fixed-point free function g is well understood; in particular, it is known that a set can compute such a function iff it can compute a diagonally nonrecursive (DNR) function g , that is, a function g such that for all e the value $\varphi_e(e)$, if defined, differs from $g(e)$ [6].

By a celebrated result of Schnorr, a set is Martin-Löf random if and only if the length n prefixes of its characteristic sequence have prefix-free Kolmogorov complexity H of at least $n - c$ for some constant c . From this equivalence, we obtain easily a proof for Kučera's result [4, Corollary 1 to Theorem 6] that any Martin-Löf random set R computes a DNR function, and hence also computes a fixed-point free function; simply let $f(e) = R(0)R(1) \dots R(e - 1)$ (where the prefixes of R are interpreted as binary expansions of natural numbers), then for appropriate constants c and c' and for all e , the prefix-free Kolmogorov complexity of the function values $\varphi_e(e)$ and $f(e)$ is at most $2 \log e + c'$ and at least $e - c$, respectively, hence for almost all e these two values differ, that is, changing f at at most finitely many places yields a DNR function. Similarly, if one lets $g(e)$ be an index for the constant function with value $f(e)$, then φ_e and $\varphi_{g(e)}$ differ at place e for almost all e , and thus a finite variant of g is fixed-point free.

We will call a set A complex if there is a nondecreasing and unbounded recursive function h such that the length $h(n)$ prefix of A has plain Kolmogorov complexity of at least n and autocomplex is defined likewise with recursive replaced by A -recursive. Obviously, any autocomplex set computes a function f such the plain Kolmogorov complexity of $f(n)$ is at least n and as observed in Proposition 3, this implication is in fact an equivalence; a similar equivalence is stated in Proposition 4 for complex sets and for computing such a function f in truth-table or weak truth-table style. By an argument similar to the one given in the last paragraph, any autocomplex set computes a DNR function. In fact, Theorem 5 asserts that the reverse implication holds too, that is, the class of autocomplex sets coincides with the class of sets that compute a DNR function, or, equivalently, with the class of sets that compute fixed-point free functions. This means that the sets relative to which the Recursion Theorem does not hold can be characterized as the autocomplex sets, that is, like for Schnorr's celebrated characterization of Martin-Löf random sets as the sets with incompressible prefixes, we obtain a characterization of a fundamental concept of theoretical computer science in terms of Kolmogorov complexity. By similar arguments, the class of complex sets coincides with the class of sets that compute a DNR function via a truth-table or weak truth-table reduction, where the DNR function then automatically is recursively bounded.

From the mentioned results on complex sets and by work of Ambos-Spies, Kjos-Hanssen, Lempp and Slaman [2], we obtain in Proposition 7 that there is

a complex set that does not compute a Martin-Löf random set, thus partially answering an open problem by Reimann [13] about extracting randomness.

Theorem 8 states that recursively enumerable (r.e.) sets are complete if and only if they are autocomplex, and are wtt-complete if and only if they are complex. Arslanov's completeness criteria in terms of DNR functions are then immediate from Theorems 5 and 8, which in summary yields simplified proofs for these criteria.

Theorem 10 asserts that the complex sets can be characterized as the sets that are not wtt-reducible to a hyperimmune set. Miller [8] demonstrated that the latter property characterizes the hyperavoidable sets, thus we obtain as corollary that a set is complex if and only if it is hyperavoidable.

In the characterization of the autocomplex sets as the sets that compute a function f such that the complexity of $f(n)$ is at least n , the values $f(n)$ of such a function f at place n might be very large and hence the complexity of the function value $f(n)$ might be arbitrarily small compared to its length. Theorem 14 states that the sets that allow to compute a function f such that the length and the plain Kolmogorov complexity of $f(n)$ are both equal to n are just the PA-complete sets and that, furthermore, these two identical classes of sets coincide with the class of sets that compute a lower bound b on plain Kolmogorov complexity such that b attains values strictly smaller than n not more often than $2^n - 1$ times. Recall in this connection that by definition a set is PA-complete if and only if it computes a complete extension of Peano arithmetic and that the concept of PA-completeness is well understood and allows several interesting characterizations, for example, a set is PA-complete if and only if it computes a $\{0, 1\}$ -valued DNR function [6].

For a word $f(n)$ of length n , in general plain Kolmogorov complexity n is not maximum, but just maximum up to an additive constant. In Theorem 15 it is demonstrated that the class of sets that allow to compute a function f such that $f(n)$ has indeed maximum plain Kolmogorov complexity depends on the choice of the universal machine used for defining plain Kolmogorov complexity; more precisely, for any recursively enumerable set B there is a universal machine such that this class coincides with the class of sets that are PA-complete and compute B . In contrast to this, Theorem 17 asserts that in the case of prefix-free Kolmogorov complexity exactly the sets that compute the halting problem allow to compute a function f such that $f(n)$ has length n and has maximum complexity among all words of the same length.

In the remainder of this section we describe some notation and review some standard concepts. If not explicitly stated differently, a set is always a subset of the natural numbers \mathbb{N} . Natural numbers are identified with binary words in the usual way, hence we can for example talk of the length $|n|$ of a number n . A set A will be identified with its characteristic sequence $A(0)A(1)\dots$, where $A(i)$ is 1 iff i is in A and $A(i)$ is 0, otherwise; this way for example we can speak of the length n prefix $A \upharpoonright n$ of a set A , which consists just of the first n bits of the characteristic sequence of A .

We write φ_e for the partial recursive functional computed by the $(e + 1)$ st Turing machine in some standard effective enumeration of all Turing machines.

Similarly, φ_e^X denotes the partial function computed by the $(e + 1)$ st oracle Turing machine on oracle X .

Recall that a set A is weak truth-table reducible (wtt-reducible) to a set B if A is computed with oracle B by some Turing machine M such that for some recursive function g , machine M will access on input n at most the first $g(n)$ bits of its oracle and that A is truth-table reducible (tt-reducible) to B if A is computed with oracle B by some Turing machine which is total for all oracles.

A function f is called fixed-point free iff $\varphi_x \neq \varphi_{f(x)}$ for all x . The partial recursive function $x \mapsto \varphi_x(x)$ is called the diagonal function and a function g is called diagonally nonrecursive (DNR) iff g is total and differs from the diagonal function at all places where the latter is defined.

We write $C(x)$ and $H(x)$ for the plain and the prefix-free Kolmogorov complexity of x , see Li and Vitányi [7] (who write “K” instead of “H”).

2 Autocomplex and complex sets

Definition 1 (Schnorr) *A function $g: \mathbb{N} \rightarrow \mathbb{N}$ is an ORDER if g is nondecreasing and unbounded.*

Definition 2 *A set A is COMPLEX if there is a recursive order g such that for all n , we have $C(A \upharpoonright n) \geq g(n)$.*

A set A is AUTOCOMPLEX if there is an A -recursive order g such that for all n , we have $C(A \upharpoonright n) \geq g(n)$.

The concepts complex and autocomplex remained the same if one would replace in their definitions plain Kolmogorov complexity C by its prefix-free variant H , and similarly the following Propositions 3 and 4 remain valid with C replaced by H . The reason is that the two variants of Kolmogorov complexity differ by less than a multiplicative constant.

Proposition 3 *For any set A , the following conditions are equivalent.*

- (1) *The set A is autocomplex.*
- (2) *There is an A -recursive function h such that for all n , $C(A \upharpoonright h(n)) \geq n$.*
- (3) *There is an A -recursive function f such that for all n , $C(f(n)) \geq n$.*

Proof. We show $(1) \Rightarrow (2) \Rightarrow (3) \Rightarrow (1)$. Given an autocomplex set A , choose an A -recursive order g where $C(A \upharpoonright n) \geq g(n)$ and in order to obtain a function h as required by (2), let

$$h(n) = \min\{l: g(l) \geq n\}.$$

Given a function h as in (2), in order to obtain a function f as required by (3), simply let $f(n)$ be equal to (an appropriate encoding of) the prefix of A of length $h(n)$. Finally, given an A -recursive function f as in (3), let $u(n)$ be an A -recursive order such that some fixed oracle Turing machine M computes f with oracle A such that M queries on input n only bits $A(m)$ of A where $m \leq u(n)$. Then for any $l \geq u(n)$, the value of $f(n)$ can be computed from n and $A \upharpoonright l$, hence

$$n \leq C(f(n)) \leq^+ C(A \upharpoonright l) + 2 \log n,$$

and thus for almost all n and all $l \geq u(n)$, we have $n/2 \leq C(A \upharpoonright l)$. As a consequence, a finite variation of the A -recursive order

$$g: n \mapsto \max\{l: u(l) \leq n\}/2$$

witnesses that A is autocomplex. \square

Proposition 4 *For any set A , the following conditions are equivalent.*

- (1) *The set A is complex.*
- (2) *There is a recursive function h such that for all n , $C(A \upharpoonright h(n)) \geq n$.*
- (3) *The set A tt-computes a function f such that for all n , $C(f(n)) \geq n$.*
- (4) *The set A wtt-computes a function f such that for all n , $C(f(n)) \geq n$.*

We omit the proof of Proposition 4, which is very similar to the proof of Proposition 3, where now when proving the implication (4) \Rightarrow (1) one exploits that the use of f is bounded by a recursive function.

Theorem 5 *A set is autocomplex if and only if it computes a DNR function.*

Proof. First assume that A is autocomplex and choose an A -recursive function f as in assertion (3) of Proposition 3. Then we have for some constant c and almost all n ,

$$C(\varphi_n(n)) \leq C(n) + c \leq \log n + 2c < n \leq C(f(n)),$$

and consequently some finite variation of f is a DNR-function.

Next suppose that A is not autocomplex. Assume for a contradiction that A computes a DNR function, that is, for some r , φ_r^A is DNR. For any z there is an index $e(z)$ such that on every input x , $\varphi_{e(z)}(x)$ is computed as follows: first, assuming that z is a code for a prefix w of an oracle (meant to be equal to A), try to decode this prefix by simulating the universal Turing machine used to define C on input z ; on success, simulate $\varphi_r(x)$ with the prefix w as oracle; if the latter computation converges with the given prefix, output the computed value.

Now consider the A -recursive function h , where $h(n)$ is the maximum of the uses of all computations of values $\varphi_r^A(e(z))$ with $|z| < n$ on oracle A . Because A is not autocomplex and by Proposition 3, there are infinitely many n such that the complexity of the length $h(n)$ prefix of A is less than n , say, witnessed by a code z_n . Then for all such n and z_n , we have

$$\varphi_r^A(e(z_n)) = \varphi_{e(z_n)}(e(z_n)),$$

hence the function computed from A by φ_r is not DNR, which contradicts our assumption. \square

The complex sets are just the sets that wtt-compute a DNR function.

Theorem 6 *For a set A the following statements are equivalent.*

- (1) *The set A is complex.*
- (2) *The set A tt-computes a DNR function.*
- (3) *The set A wtt-computes a DNR function.*

In particular, the sets that permit to wtt-compute DNR-functions and to wtt-compute recursively bounded DNR-functions coincide.

Proof. (1) implies (2): Similar to the proof of Theorem 5, the set A tt-computes a DNR function where $f(n)$ is equal to the prefix of A of length $l(n)$, where l is defined as above but can now be chosen to be recursive.

(2) implies (3): This is immediate from the definition.

(3) implies (1): Follows as in the proof of Theorem 5, where now the function h can be chosen to be recursive.

The concluding remark follows because a DNR function as in (2) is bounded by the maximum of the function values over all possible oracles. \square

The constructive Hausdorff dimension of a set A can be defined as the limit inferior of the relative Kolmogorov complexities $C(A \upharpoonright n)/n$ of the prefixes of A . Reimann [13] has asked whether one can “extract randomness” in the sense that any set with constructive Hausdorff dimension $\alpha > 0$ computes a set with constructive Hausdorff dimension 1 or even a Martin-Löf random set. By Theorem 6 and work of Ambos-Spies, Kjos-Hanssen, Lempp and Slaman [2] (which is in turn based on an unpublished construction by Kumabe), we obtain as a partial answer to this question that there is a complex set that does not compute a Martin-Löf random set, that is, in general it is not possible to compute a set of roughly maximum complexity from a set that has a certain minimum and effectively specified amount of randomness. Reimann and Slaman [14] have independently announced a proof of Theorem 7 by means of a direct construction.

Theorem 7 *There is a complex set that does not compute a Martin-Löf random set.*

Proof. Ambos-Spies, Kjos-Hanssen, Lempp and Slaman [2] demonstrate that for any recursive function h there is a DNR function d that cannot compute a DNR function that is bounded by $h(n)$. In their construction it is implicit that the constructed function d can be made recursively bounded, hence the function d is tt-reducible to its graph D and, by Theorem 6, the set D is complex. Apply this construction to the function $h(n) = 2^n$ and assume that the graph D of the resulting function d computes a Martin-Löf random set R . Then D also computes the function $f: n \mapsto R(0) \dots R(n)$, which is $h(n)$ -bounded. Now $f(n)$ has H-complexity of at least n for almost all n , hence $f(n)$ differs from $\varphi_n(n)$ for almost all n , that is, by changing f at finitely many places, we obtain an $h(n)$ -bounded DNR function recursive in D , contradicting the choice of D . \square

Theorem 8 *An r.e. set is Turing complete if and only if it is autocomplex. An r.e. set is wtt-complete if and only if it is complex.*

Proof. Fix any r.e. set A and let A_s be the finite set of all elements that have been enumerated after s steps of some fixed effective enumeration of A ; similarly, fix finite approximations K_s to the halting problem K . We demonstrate the slightly less involved second equivalence before the first one.

First assume that A is wtt-complete. That is, A wtt-computes the halting problem, which in turn wtt-computes the function f that maps n to the least

word that has C-complexity of at least n . Since wtt-reductions compose, A wtt-computes f , hence A is complex by Proposition 4. Conversely, if the set A is complex, by Proposition 4 fix a recursive function h such that for all n we have $C(A \upharpoonright h(n)) \geq n$. For all n , let

$$k(n) = \min\{s \in \mathbb{N} : K(n) = K_s(n)\}, \quad a(n) = \min\{s \in \mathbb{N} : A \upharpoonright h(n) = A_s \upharpoonright h(n)\}.$$

Obviously A wtt-computes a , hence in order to show that A wtt-computes K it suffices to show that for almost all n we have $k(n) \leq a(n)$. For a proof by contradiction, assume that there are infinitely many n such that $k(n) > a(n)$. Each such n must be in K because otherwise $k(n)$ were equal to 0; consequently, given any such n , the word

$$w_n = A \upharpoonright h(n) = A_{a(n)} \upharpoonright h(n) = A_{k(n)} \upharpoonright h(n)$$

can be computed as follows: first compute $k(n)$ by simulating the given approximation to K , then compute $h(n)$ and let w_n be equal to $A_{k(n)} \upharpoonright h(n)$. Consequently, up to an additive constant we have $C(w_n) \leq C(n)$, that is, $C(w_n)$ is in $O(\log n)$, which contradicts the choice of w_n for all sufficiently large n .

The proof that Turing complete sets are autocomplex is almost literally the same as for the corresponding assertion for wtt-complete sets and is omitted. In order to prove the reverse implication, assume that A is autocomplex and according to Proposition 3, fix an A -recursive function h such that for all n we have $C(A \upharpoonright h(n)) \geq n$. Let $a(n)$ be equal to the least s such that A_s agrees with A on all elements that are queried while computing $h(n)$ and on all numbers up to $h(n)$. Like in the case of wtt-complete sets, we can argue that A computes a and that for all n where $k(n) > a(n)$, the word $A \upharpoonright h(n)$ can be computed from n , hence for almost all n we have $k(n) \leq a(n)$ and A computes K . \square

As immediate corollaries to Theorems 5, 6, and 8, we obtain the following well-known characterizations of T- and wtt-completeness [11, Theorem III.1.5 and Proposition III.8.17], where the latter characterization is known as Arslanov's completeness criterion [1, 6].

Corollary 9 *An r.e. set is Turing complete if and only if it computes a DNR-function. An r.e. set is wtt-complete if and only if it wtt-computes a DNR-function.*

A set $A = \{a_0, a_1, \dots\}$ with $a_0 < a_1 < \dots$ is called hyperimmune if A is infinite and there is no recursive function h such that $a_n \leq h(n)$ for all n [11]. A further characterization is that there is no recursive function f such that A intersects almost all sets of the form $\{n+1, n+2, \dots, f(n)\}$. Intuitively speaking, hyperimmune sets have large gaps that exceed all recursive bounds, hence if a set A is wtt-reducible to a hyperimmune set, that is, is reducible by a reduction with recursively bounded use, then A must have prefixes of very low complexity. This intuition is made precise by the next proposition which gives another characterization of complex sets.

Theorem 10 *A set is complex if and only if it is not wtt-reducible to a hyperimmune set.*

Proof. First assume that f, g are recursive functions, A is wtt-reducible to a hyperimmune set $B = \{b_0, b_1, \dots\}$ with use f and $C(A(0)A(1)\dots A(g(n))) \geq n$ for all n . There are infinitely many n such that $f(g(4^{b_n})) < b_{n+1}$. Thus there is a constant c with $C(A(0)A(1)\dots A(g(4^{b_n}))) \leq 2^{b_n+c}$ for infinitely many n ; this happens at those n where at the computation of $A(0)A(1)\dots A(g(4^{b_n}))$ relative to B only the characteristic function of B up to b_n has to be taken into account since it is 0 afterwards up to the query-bound $f(g(4^{b_n}))$. On the other hand, $C(A(0)A(1)\dots A(g(4^{b_n}))) \geq 4^{b_n}$ for all n . This gives $4^{b_n} \leq 2^{b_n+c}$ for infinitely many n and contradicts to the fact that the sequence b_0, b_1, \dots is strictly increasing as it is the ascending sequence of elements of an infinite set. Thus A is not complex.

Next assume that A is not wtt-reducible to any hyperimmune set. Let $p(m)$ be that word σ such that 1σ has the binary value $m+1$, so $p(0) = \lambda$, $p(1) = 0$, $p(2) = 1$, $p(3) = 00$ and so on. Let U be the universal machine on which C is based and assume that U is such that $C(A(0)A(1)\dots A(n)) \leq n+1$ for all n . Now let $f(n)$ be the first m such that $U(p(m))$ is a word extending $A(0)A(1)\dots A(n)$ and let

$$B = \{(n, m) : f(n) = m \wedge \forall k < n (f(k) \neq m)\}.$$

Now $A \leq_{wtt} B$ since $A(n) = U(p(m))(n)$ for the maximal m such that $(k, m) \in B \wedge k \leq n$; one can find this m by querying B at (i, j) for all $(i, j) \in \{0, 1, \dots, n\} \times \{0, 1, \dots, 2^{n+1}\}$. Therefore B is not hyperimmune and there is a recursive function g such that B has more than 2^{n+1} elements falling into the rectangle $\{0, 1, \dots, h(n)\} \times \{0, 1, \dots, g(n)\}$. Now one knows that $f(A(0)A(1)\dots A(g(n))) \geq 2^{n+1}$ and thus $C(A(0)A(1)\dots A(g(n))) \geq n$. So A is complex. \square

Remark 11 Post [12] introduced the notion of hyperimmune sets and demonstrated that every r.e. set with a hyperimmune complement is wtt-incomplete and that there are such sets, that is, hyperimmunity was used as a tool for constructing an r.e. wtt-incomplete set. Moreover, Post showed that if an r.e. set A is wtt-complete then A is not wtt-reducible to any hyperimmune set. This implication is in fact an equivalence, that is, an r.e. set is wtt-complete if and only if it is not wtt-reducible to a hyperimmune set, as is immediate from Theorems 6 and 8.

Miller [8] introduced the notion of hyperavoidable set. A set is hyperavoidable iff it differs from any characteristic function of a recursive set on a prefix of length computable from an index for that recursive set. Formally, a set A is hyperavoidable if there is a nondecreasing and unbounded recursive function h such that for all e , whenever $\varphi_e(x)$ is defined for all $y < h(e)$, then we have

$$A(0) \dots A(h(e) - 1) \neq \varphi_e(0) \dots \varphi_e(h(e) - 1).$$

Among other results on hyperavoidable sets, Miller showed that hyperavoidability can be characterized by not being wtt-reducible to any hyperimmune set.

Theorem 12 [8, Theorem 4.6.4] *A set is hyperavoidable if and only if it is not wtt-reducible to any hyperimmune set.*

The following proposition is then immediate from Theorems 10 and 12.

Proposition 13 *A set is hyperavoidable if and only if it is complex.*

3 Plain Kolmogorov complexity and completions of Peano arithmetic

By Propositions 3 and Theorem 5, computing a DNR function is equivalent to the ability to compute on input n a word $f(n)$ of C-complexity of at least n . The next theorem shows that if one enforces the additional constraint that the word $f(n)$ has length n , that is, if one requires $f(n)$ to be an incompressible word of length n , then one obtains a characterization of the strictly smaller class of PA-complete sets. Recall that a set A is PA-complete if and only if A computes a DNR function with finite range, which by a result of Jockusch [6] in turn is equivalent to computing a $\{0, 1\}$ -valued DNR function.

Theorem 14 *The following is equivalent for every set A .*

- (1) *A computes a $\{0, 1\}$ -valued DNR function.*
- (2) *A computes a function f such that for all n , $f(n)$ has length n and satisfies $C(f(n)) \geq n$.*
- (3) *A computes a lower bound b on plain complexity C such that for all n there are at most $2^n - 1$ many x with $b(x) < n$.*

Proof. (3) implies (2): Just let $f(n)$ be the lexicographically first word y of length n such that $b(y) \geq n$. This word exists by the condition that there are at most $2^n - 1$ words x with $b(x) < n$. Since b is a lower bound for C , one has that $C(f(n)) \geq n$ for all n . Furthermore, f is computed from b .

(2) implies (1): Let the partial recursive function ψ be defined by $\psi(x) = x\varphi_n(n)$ where n is the length of x and ψ is defined if and only if $\varphi_n(n)$ is defined. Then there is a constant c such that $C(\psi(x)) < n + c$ for all x, n with $x \in \{0, 1\}^n$. In order to obtain a DNR function d with finite domain that is computed by f , let $d(n)$ consists of the last c bits of $f(n + c)$. Now consider any n such that $\varphi_n(n)$ is defined. If we let x be the first n bits of $f(n + c)$, then we have

$$xd(n) = f(n + c) \neq \psi(x) = x\varphi_n(n)$$

where the inequality holds by assumption on f and because of $C(\psi(x)) < n + c$. Thus d is a DNR function and its range is the finite set $\{0, 1\}^c$. By the already mentioned result of Jockusch [6], this implies that f computes a $\{0, 1\}$ -valued DNR function.

(1) implies (3): Assume that A computes a $\{0, 1\}$ -valued DNR function and that hence A is PA-complete. Consider the Π_1^0 class of all sets G that satisfy the following two conditions:

- $\forall p, x, s (U_s(p) \downarrow = x \Rightarrow (p, x) \in G)$;
- $\forall p, x, y ((p, x) \in G \wedge (p, y) \in G \Rightarrow x = y)$.

Since A is PA-complete, by the Scott Basis Theorem (see Odifreddi [11, Theorem V.5.35]) this Π_1^0 class contains a set that is computable in A ; fix such a set G . Now one defines $b(x) = \min\{|p| : |p| < n \ \& \ (p, x) \in G\}$. By the first condition, the function b is a lower bound for C . By the second condition, any word p can occur in at most one pair (p, x) in G , hence there are at most $2^n - 1$ many x where there is such a pair with $|p| < n$, or equivalently, where $b(x) < n$. \square

One might ask whether one can strengthen Theorem 14 such that any PA-complete set A , that is, any set that satisfies the first condition in the theorem, computes a function f such $f(n)$ is a word of length n that, instead of just being incompressible as required by the second condition, has maximum plain Kolmogorov complexity among all words of the same length. The answer to this question depends on the universal machine that is used to define plain complexity; more precisely, for every r.e. oracle B one can compute a corresponding universal machine which makes this problem hard not only for PA but also for B . The proof of Theorem 15 is omitted due to space considerations.

Theorem 15 *For every recursively enumerable oracle B there is a universal machine U_B such that the following two conditions are equivalent for every oracle A :*

- (1) A has PA-complete degree and $A \geq_T B$.
- (2) There is a function $f \leq_T A$ such that for all n and for all $x \in \{0,1\}^n$, $f(n) \in \{0,1\}^n$ and $C_B(f(n)) \geq C(x)$, where C_B is the plain Kolmogorov complexity based on the universal machine U_B .

4 Computing words with maximum prefix-free Kolmogorov complexity

While PA-completeness can be characterized in terms of C , an analogous result for H fails. First, one cannot replace C -incompressibility by H -incompressibility since relative to any Martin-Löf random set A , which includes certain non-PA-complete sets, one can compute the function mapping n to the H -incompressible word $A(0)\dots A(n)$. So one might consider the sets A which permit to compute words of maximal complexity in order to obtain a characterization of PA-complete sets. However, instead the corresponding notion gives a characterization of the sets that compute the halting problem K . The proof of this result makes use of the following proposition, which we state without proof because lack of space. Note that the following proposition does not hold with C in place of H .

Proposition 16 *Let f be a partial recursive surjective function with $|f(x)| < |x|$ for all x in the domain of f . Then there is a constant c and an enumeration U_s of the universal prefix-free machine U such that*

$$\forall n \forall x \forall s (f(x) = n \wedge H_s(x) = H(x) \Rightarrow H_s(n) \leq H(n) + c)$$

where H_s is the approximation to H based on U_s .

In contrast to Theorem 15, the following result does not depend on the choice of the universal machine that is used when defining Kolmogorov complexity.

Theorem 17 *A set A computes the halting problem K if and only if there is a function $f \leq_T A$ such that for all n , the word $f(n)$ has length n and has maximum H -complexity among all words of the same length, that is, $H(x) \leq H(f(n))$ for all words x of length n .*

Proof. Since H is K -recursive, a function f as required can obviously be computed if $A \geq_T K$. For the reverse direction, assume that $f \leq_T A$ is a function as stated in the theorem. Let U be the universal prefix-free machine on which H is based. Given any n, m and any $o < 2^m$, let h map every word of length $2^{n+m+1} + 2^m + o$ to n ; h is undefined on words of length $0, 1, 2$. Note that $|h(x)| < |x|$ for all x . By Proposition 16 there is an enumeration U_s of U and a constant, here called c_1 , such that

$$\forall n \forall x \forall s (h(x) = n \wedge H_s(x) = H(x) \Rightarrow H_s(n) \leq H(n) + c_1).$$

Let P_0, P_1, \dots be an enumeration of all primitive-recursive partitions of the natural numbers and let $P_{m,0}, P_{m,1}, \dots$ be the members of partition P_m enumerated such that the member $P_{m,o}$ exists whenever $P_{m,o+1}$ exists. We can assume that every partition in the enumeration has infinitely many indices. Now define a family of partial recursive function T_0, T_1, \dots such that each T_k on input p does the following steps:

- Compute the first s such that $U_s(p)$ is defined;
- If $U_s(p)$ is defined then check whether there are values n, m, ℓ, x, y, z such that $U(p) = xz$, $|x| = 2^{n+m+1} + 2^m$, $\ell = 2^{2^{|z|}}$ and $\ell < 2^m$.
- If this also goes through and $H_s(n) \geq k$ then search for o such that $H_s(n) - k \in P_{m,o}$.
- If all previous steps have gone through and $o < 2^\ell$ then let $T_k(p) = xy$ for the unique $y \in \{0, 1\}^\ell$ with $bv(y) = o$.

Here $bv(y)$ is the binary value of y , for example, $bv(00101) = 5$. The machines T_k are prefix-free and there is a constant c_2 such that for all $k \leq c_1$, $H(T_k(p)) \leq |p| + c_2$. Furthermore, there is a constant c_3 such that $H(F(o + c_3)) + c_2 < H(F(o + 2^{2^{c_3}}))$ for all o . In particular one has

$$\forall n \forall m > c_3 + 2 \forall p (|U(p)| = 2^{n+m+1} + 2^m + c_3 \Rightarrow \forall k \leq c_1 (H(T_k(p)) < H(F(2^{n+m+1} + 2^m + c_4))))).$$

where $c_4 = 2^{2^{c_3}}$. Given any n and $m > 2^{c_3}$, let y be the last c_4 bits of $F(2^{n+m+1} + 2^m + c_4)$. Then $P_{m,bv(y)}$ does either not exist or not contain $H(n)$.

Thus one can run the following A -recursive algorithm to determine for any given n a set of up to $2^{c_4} - 1$ elements which contains $H(n)$ by the following algorithm.

- Let $E = \{0, 1, \dots, 2n + 2\}$ and $m = c_4 + 1$.
- While $|E| \geq 2^{c_4}$ Do Begin $m = m + 1$,
Determine the word y consisting of the last c_4 bits of $f(2^{n+m+1} + 2^m + c_4)$,
If $P_{m,bv(y)}$ exists and intersects E then let $E = E - P_{m,bv(y)}$ End.
- Output E .

This algorithm terminates since whenever $|E| \geq 2^{c_4}$ at some stage m then there is $o > m$ such that P_o has 2^{c_4} members all intersecting E and one of them will be removed so that E loses an element in one of the stages $m + 1, \dots, o$. Thus the above algorithm computes relative to A for input n a set of up to $2^{c_4} - 1$ elements containing $H(n)$. By a result of Beigel, Buhrman, Fejer, Fortnow, Grabowski, Longpré, Muchnik, Stephan and Torenvliet [5], such an A -recursive algorithm can only exist if $K \leq_T A$. \square

Remark 18 Theorem 17 and its proof answer a question of Calude, who had asked whether the statement of the theorem is true with the condition that $H(f(n))$ is maximum (among all words of the same length) replaced by the condition that $H(f(n))$ is maximum up to an additive constant. This variant of Theorem 17 can be obtained by a minor adaptation of the proof of the theorem given above.

Nies [10] pointed out that for any n the word x of maximum prefix-free Kolmogorov complexity among all words of length n satisfies, up to an additive constant, the equality $H(x) = n + H(n)$, hence the variant of Theorem 17 can be rephrased as follows. For any oracle A , $A \geq_T K$ if and only if there is a function $f \leq_T A$ and a constant c such that for all n , $f(n) \in \{0, 1\}^n$ and $H(f(n)) \geq f(n) + H(f(n)) - c$.

Acknowledgements We would like to thank Cristian Calude and André Nies for helpful discussion.

References

1. Marat M. Arslanov, On some generalizations of the Fixed-Point Theorem, *Soviet Mathematics (Iz. VUZ)*, Russian, 25(5):9–16, 1981, English translation, 25(5):1–10, 1981.
2. Klaus Ambos-Spies, Bjørn Kjos-Hanssen, Steffen Lempp and Theodore A. Slaman. Comparing DNR and WWKL, *Journal of Symbolic Logic*, 69:1089–1104, 2004.
3. Cristian S. Calude. Private Communication, 2005.
4. Antonín Kučera. Measure, Π_1^0 -classes and complete extensions of PA. In *Recursion theory week 1984, Lecture Notes in Mathematics* 1141:245–259, 1985.
5. Richard Beigel, Harry Buhrman, Peter Fejer, Lance Fortnow, Piotr Grabowski, Luc Longpré, Andrej Muchnik, Frank Stephan, and Leen Torenvliet. Enumerations of the Kolmogorov function. *Electronic Colloquium on Computational Complexity*, TR04-015, 2004.
6. Carl G. Jockusch, Jr. Degrees of functions with no fixed points. In *Logic, methodology and philosophy of science, VIII (Moscow, 1987)*, volume 126 of *Stud. Logic Found. Math.*, pages 191–201. North-Holland, Amsterdam, 1989.
7. Ming Li and Paul Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Graduate texts in Computer Science, Springer, Heidelberg, 1997.
8. Joseph Stephen Miller. Π_1^0 classes in Computable Analysis and Topology. PhD thesis, Cornell University, 2002.
9. Bjørn Kjos-Hanssen, André Nies and Frank Stephan. Lowness for the class of Schnorr random reals. *SIAM Journal on Computing*, to appear.
10. André Nies. Private Communication, 2004.
11. Piergiorgio Odifreddi. *Classical Recursion Theory*. North-Holland, Amsterdam, 1989.
12. Emil Post. Recursively enumerable sets of positive integers and their decision problems, *Bulletin of the American Mathematical Society*, 50:284–316, 1944.
13. Jan Reimann. *Computability and Fractal Dimension*. Doctoral dissertation, Fakultät für Mathematik und Informatik, Universität Heidelberg, INF 288, D-69120 Heidelberg, Germany, 2004.
14. Jan Reimann. Extracting randomness from sequences of positive dimension. Post of an open problem in the recursion theory section of the Mathematical Logic Forum at math.berkeley.edu/Logic/problems/, 2004.