Math. Struct. in Comp. Science (2009), *vol.* 19, *pp.* 45–57.
doi:10.1017/S0960129508007238 Printed in the United Kingdom

On universal computably enumerable prefix codes

CRISTIAN S. CALUDE[†] and LUDWIG STAIGER[‡]

[†]*Department of Computer Science, The University of Auckland, Private Bag 92019, Auckland, New Zealand*
Email: cristian@cs.auckland.ac.nz

[‡]*Martin-Luther-Universität Halle-Wittenberg, Institut für Informatik, D - 06099 Halle, Germany*
Email: staiger@informatik.uni-halle.de

Received 11 October 2007

We study computably enumerable (c.e.) prefix codes that are capable of coding all positive integers in an optimal way up to a fixed constant: these codes will be called universal. We prove various characterisations of these codes, including the following one: a c.e. prefix code is universal if and only if it contains the domain of a universal self-delimiting Turing machine. Finally, we study various properties of these codes from the points of view of computability, maximality and density.

1. Introduction and notation

We study computably enumerable prefix codes that are capable of coding all positive integers in an optimal way up to a fixed constant: these codes will be called universal. Our arguments combine elementary facts from coding theory, algorithmic information theory and formal language theory. We prove various characterisations of these codes including the following one: a c.e. prefix code is universal if and only if it contains the domain of a universal self-delimiting Turing machine. Various properties of these codes are then presented.

We will follow the notation in Calude (2002). We use $\mathbb{N} = \{0, 1, 2, \ldots\}$ to denote the set of positive integers. The cardinality of a set A is denoted by $|A|$. Let us fix $X = \{0, \ldots, r-1\}$ an alphabet of cardinality r, and use X^* to denote the set of finite strings (words) on X, including the *empty* string λ.

The length of the string w is denoted by $|w|$, and we use $X^i = \{w \in X^* \mid |w| = i\}$, $X^{\leqslant i} = \{w \in X^* \mid |w| \leqslant i\}$ and $X^{\geqslant i} = \{w \in X^* \mid |w| \geqslant i\}$ to denote the sets of strings having lengths exactly i, not larger than i, or not smaller than i, respectively. If v is a prefix of w, we write $v \sqsubseteq w$, and write $v \sqsubset w$ if $v \sqsubseteq w$ and $v \neq w$. A natural ordering of X^* is the *quasi-lexicographical* (or *length-lexicographical*) ordering '$\leqslant_{\mathrm{qlex}}$' where strings are ordered first according to their length, and strings of the same length are then ordered lexicographically (with respect to some ordering of the alphabet X)[†]. We use $\mathsf{string}_r(n)$ to denote the

[†] Work done in Halle; the support of Martin-Luther University and Institute of Informatics is gratefully acknowledged. Part of this project was supported by UARC Grant 3607895/2006.

[†] This ordering is not to be confused with the lexicographical ordering where the string 1 is preceded by all strings starting with 0.

nth string in the quasi-lexicographical ordering of $X^* = \{0, \ldots, r-1\}^*$, for example, $\mathsf{string}_r(0) = \lambda$, $\mathsf{string}_r(1) = 0$, $\mathsf{string}_r(2) = 1, \ldots, \mathsf{string}_r(r+1) = 00, \ldots$, and so on.

Moreover, we fix a prefix-free encoding of strings in X^* in the same way as, for example, in Zvonkin and Levin (1970), so that for $w = x_1 \cdots x_l$ where $x_i \in X$, $l \geqslant 0$ we set $\overline{x_1 \cdots x_l} := 0x_1 0x_2 \cdots 0x_l 1$.

For $V, W \subseteq X^*$, we use VW to denote the set $\{vw \mid v \in V \wedge \in W\}$ of concatenations of strings from V with strings from W. For $V = \{u\}$ we write uW instead of $\{u\}W$. A *prefix code* is a prefix-free subset of strings. Prefix codes over X satisfy Kraft's inequality: $\sum_{w \in A} r^{-|w|} \leqslant 1$.

A *self-delimiting Turing machine* (a *machine* for short) is a Turing machine C processing binary strings such that its program set (domain) $\mathrm{dom}(C) = \{\pi \mid \pi \in X^* \wedge C(\pi) \text{ halts}\}$ is a prefix-free set of strings. As usual, we define the *self-delimiting (prefix, or program-size) complexity* of a string w with respect to a machine C as $H_C(w) := \inf\{|\pi| \mid \pi \in X^* \wedge C(\pi) = w\}$. See Chaitin (1987), Calude (2002) and Downey and Hirschfeldt (to appear) for further details.

A prefix code is *computably enumerable* (c.e.) if and only if it is the domain of a self-delimiting Turing machine.

We can effectively construct a machine U (called *universal*) such that for every machine C, there exists a constant k (depending only on U and C) such that for every string $\pi \in \mathrm{dom}(C)$ there exists a string $\pi' \in \mathrm{dom}(U)$ such that $U(\pi') = C(\pi)$ and $|\pi'| \leqslant |\pi| + k$. A *prefix-universal* machine U is a special universal machine defined by the following property: for every self-delimiting Turing machine C there exists a string w (depending only on U and C) such that for every string $\pi \in \mathrm{dom}(C)$ we have $U(w\pi) = C(\pi)$. We can effectively construct prefix-universal machines; there exist universal machines that are not prefix-universal. All quantifiers in the definition of universality and prefix-universality are *effective*.

2. Motivation

Consider the binary alphabet $X = \{0, 1\}$. The computable prefix code $S = \{1^n 0 : n \geqslant 0\}$ codes every integer $n \geqslant 0$ with a string of $n+1$ bits. A better solution is given by the computable prefix code $S = \{1^{\log n} 0 \mathsf{string}_2(n) : n \geqslant 0\}$, which codes every integer $n \geqslant 0$ with a string of $2\log n + 1$ bits. An even better solution is a computable prefix code T that codes every integer $n \geqslant 0$ with a string of length $\log n + 2\log n \log n + 1$ bits. In Levenšteĭn (1968), two prefix codes for the natural numbers are introduced and shown to:

(a) have an asymptotically minimal redundancy; and

(b) be computable by a Turing machine with a minimal delay.

We may ask: is there a *best way* of representing integers with computable prefix codes, or, more generally, with c.e. prefix codes? There are various ways to define optimality; here we will focus on set-theoretic maximality, information-theoretic (rate/capacity) and computable one-to-one translations (embedability).

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

3. Properties of universal c.e. prefix codes

In this section we define and characterise universal c.e. prefix codes. We start with a theorem that characterises universal c.e. prefix codes. Then we give a non-computability result, and the final subsection is devoted to some consequences.

3.1. *A characterisation theorem*

Here we prove the following equivalences.

Theorem 1. Let $V \subseteq X^*$ be a c.e. prefix code. Then, the following statements are equivalent:

1 There exists a universal machine U such that $V \supseteq \mathrm{dom}(U)$.

2 For every partial computable one–one function $g : \mathbb{N} \to X^*$ having a prefix-free range, there exist a partial computable one–one function $f : \mathbb{N} \to X^*$ and a constant $k \in \mathbb{N}$ such that:

(a) $f(\mathrm{dom}(f)) \subseteq V$.

(b) $\mathrm{dom}(g) \subseteq \mathrm{dom}(f)$ and $|f(n)| \leqslant |g(n)| + k$, for every $n \in \mathrm{dom}(g)$.

3 For every computable one–one function $g : \mathbb{N} \to X^*$ having a prefix-free range, there exist a computable one–one function $f : \mathbb{N} \to X^*$ and a constant $k \in \mathbb{N}$ such that:

(a) $f(\mathbb{N}) \subseteq V$.

(b) $|f(n)| \leqslant |g(n)| + k$, for every $n \in \mathbb{N}$.

4 For every c.e. prefix code $D \subseteq X^*$ there exist a partial computable one–one function $\varphi : X^* \to X^*$ and a constant $k \in \mathbb{N}$ such that:

(a) $D \subseteq \mathrm{dom}(\varphi)$, $\varphi(D) \subseteq V$.

(b) $|\varphi(u)| \leqslant |u| + k$, for every $u \in \mathrm{dom}(\varphi)$.

Proof. For the implication $1 \Rightarrow 2$ we assume that U is a universal machine and $V \supseteq \mathrm{dom}(U)$. Assume also that g is a partial computable one–one function from positive integers to strings having a prefix-free range. Define $C(g(n)) = g(n)$, for every $n \in \mathrm{dom}(g)$.

Clearly, C is a machine, so by virtue of the universality of U there exists a constant $k \in \mathbb{N}$ such that for every $n \in \mathrm{dom}(g)$ there exists a string $x_n \in \mathrm{dom}(U) \subseteq V$ such that $U(x_n) = C(g(n)) = g(n)$ and $|x_n| \leqslant |g(n)| + k$. Now, using the constant k from above, define

$$f(n) := \mu_\psi w(|w| \leqslant |g(n)| + k \wedge U(w) = g(n)), \tag{1}$$

where w is the first string satisfying the condition taken with respect to some computable enumeration ψ of $\mathrm{dom}(U)$. Clearly, f is partial computable. According to the choice of the constant k, $f(n)$ is defined whenever $g(n)$ is defined, and, moreover, in this case $U(f(n)) = g(n)$, and $|f(n)| \leqslant |g(n)| + k$, for all $n \in \mathrm{dom}(g)$. Thus, $\mathrm{dom}(f) \supseteq \mathrm{dom}(g)$ and $f(\mathrm{dom}(f)) \subseteq \mathrm{dom}(U) \subseteq V$.

For the implication $2 \Rightarrow 3$ we just observe that f is total because g is total and $\mathrm{dom}(g) \subseteq \mathrm{dom}(f)$.

If D is finite, the implication $3 \Rightarrow 4$ is trivial, just take as images of the strings $w \in D$ the first $|D|$ strings in V.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

Now let $D \subseteq X^*$ be an infinite c.e. prefix code and take a computable one–one function $g : \mathbb{N} \to D$ that enumerates D. In view of 3, there exists a constant k and a computable one–one function $f : \mathbb{N} \to X^*$ such that $f(\mathbb{N}) \subseteq V$, and $|f(n)| \leqslant |g(n)| + k$ for each n. Next define the mapping φ by $\varphi(v) = f(g^{-1}(v))$. The mapping φ is well defined (because both functions g, f are one–one) and partial computable; moreover, $\mathrm{dom}(\varphi) \supseteq g(\mathbb{N}) = D$ and $\varphi(v) \in V$, for all $v \in D$.

For every $v \in D$, we have $|\varphi(v)| = |f(g^{-1}(v))| \leqslant |g(g^{-1}(v))| + k = |v| + k$, because of condition 3.b, and $\varphi(D) \subseteq V$.

Finally, for the implication $4 \Rightarrow 1$ we consider a universal machine U' and put $D = \mathrm{dom}(U')$. In view of 4, there exist a partial computable one–one function $\varphi : X^* \to V$, and a constant k (each depending upon V, D) such that conditions 4.a, 4.b are satisfied. Define $U(u) = U'(\varphi^{-1}(u))$.

We have $\mathrm{dom}(U) = \varphi(X^*) \subseteq V$, by 4.b, a prefix code. To show that U is a universal machine, we show that $H_U(w) \leqslant H_{U'}(w) + k$ for each $w \in X^*$.

Let $w \in X^*$. Then there is a $v \in \mathrm{dom}(U')$ such that $U'(v) = w$ and $|v| = H_{U'}(w)$. Since, by definition, $w = U'(v) = U(\varphi(v))$, we have $H_U(w) \leqslant |\varphi(v)| \leqslant |v| + k = H_{U'}(w) + k$. □

For the case $V = \mathrm{dom}(U)$, since U is a universal machine, we can strengthen the condition 4 in Theorem 1 in the following way.

Corollary 2. For every c.e. prefix code $D \subseteq X^*$ and every universal machine U there are a partial computable one–one function $\varphi : X^* \to X^*$ and a constant $k \in \mathbb{N}$ such that:

(a) $D \subseteq \mathrm{dom}(\varphi)$, $\varphi(D) \subseteq \mathrm{dom}(U)$.
(b) $|\varphi(u)| \leqslant |u| + k$, for all $u \in D$.
(c) $U(\varphi(u)) = u$, for all $u \in D$.

Proof. Again the case of finite prefix codes is trivial; map $v \in D$ to a shortest $u \in \mathrm{dom}(U)$ such that $U(u) = v$.

If D is infinite, consider the implication $1 \Rightarrow 2$ of the proof of Theorem 1. If we choose $g : \mathbb{N} \to X^*$ as a function enumerating exactly the set D and define $f : \mathbb{N} \to X^*$ as in Equation (1), we get $U(f(n)) = g(n)$ and $|f(n)| \leqslant |g(n)| + k$. Now, as above, let $\varphi(u) := f(g^{-1}(n))$, and we obtain $U(\varphi(u)) = u$ and $|\varphi(u)| \leqslant |u| + k$ for $u = g(n) \in D$. □

Definition 3. We say that a c.e. prefix code is *universal* if it satisfies one of the equivalent conditions 1 – 4 in Theorem 1.

As an immediate consequence of Theorem 1.4 or Corollary 2, we obtain the following lemma.

Lemma 4. Let $V \subseteq X^*$ be a universal c.e. prefix code. Then for every c.e. prefix code $D \subseteq X^*$, there is a constant $k \in \mathbb{N}$ such that for all $l \in \mathbb{N}$, the inequality $|D \cap X^{\leqslant l}| \leqslant |V \cap X^{\leqslant l+k}|$ holds.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

For domains of prefix-universal machines U, we have the following characterisation, which is simpler than the one given in Theorem 1.

Fact 5. Let $V \subseteq X^*$ be a c.e. prefix code. The following statements are equivalent:

1 There exists a prefix-universal machine U such that $V = \mathrm{dom}(U)$.
2 For every c.e. prefix code $D \subseteq X^*$, there exists a string $w \in X^*$ such that $wD = V \cap wX^*$.

Proof. The implication $1 \Rightarrow 2$ follows from the definition of a prefix-universal machine. For the converse implication, we consider a universal machine U' and put $D = \mathrm{dom}(U')$. As D is a c.e. code, there exists a string $w \in X^*$ such that $wD = V \cap wX^*$.

We now define U by the formula:

$$U(v) = \begin{cases} U'(u) & \text{if } v = w \cdot u \\ \lambda & \text{if } w \not\sqsubseteq v \text{ and } v \in V \\ \text{undefined} & \text{otherwise.} \end{cases}$$

It is clear that U is a universal machine; if U' is prefix-universal, then so is U. □

3.2. *A non-computability result*

Although every c.e. prefix code can be in a one-to-one manner effectively embedded into any universal c.e. prefix code, it turns out that no universal c.e. prefix code is contained in a computable prefix code. To this end, we consider the language-theoretic density of (prefix) codes.

Lemma 6. If $V \subseteq X^*$ is a prefix code and $|X| = r$, then for every $l \in \mathbb{N}$ there is an $m \in \mathbb{N}$ such that $|V \cap X^{\leqslant l+m}| < r^m$.

Proof. Since $V \subseteq X^*$ satisfies Kraft's inequality $\sum_{v \in V} |X|^{-|v|} \leqslant 1$, it has density

$$\lim_{m \to \infty} \frac{|V \cap X^{\leqslant m}|}{|X|^m} = 0$$

(*cf.* Berstel and Perrin (1985)). The proof then follows immediately from this. □

Universal c.e. prefix codes have the following property.

Theorem 7 (Nies). Every universal c.e. prefix code is Turing complete.

A recursion-theoretic proof – communicated in Nies (2007) – can be found in Nies (to appear, Section 2.2).

Lemma 6 and the results of the previous section allow us to give an elementary direct proof of the weaker fact that no universal c.e. code can be computable.

Corollary 8. No universal c.e. prefix code is computable.

Before proceeding to the proof, we will briefly sketch the idea behind it. Under the assumption that the universal c.e. prefix code $V \subseteq X^*$ is computable from V we construct a computable code D such that for every $k \in \mathbb{N}$ there is an $l_k \in \mathbb{N}$ such that $|D \cap X^{\leqslant l_k}| > |V \cap X^{\leqslant l_k + k}|$. This is done by choosing a computable sequence $(v_k)_{k \in \mathbb{N}}$ of strings $v_k \in V$,

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

$|v_k| < |v_{k+1}|$, and replacing in V the string v_k by a suitably large set of strings $v_k \cdot X^{m_k}$. Then we show that D is computable if V is computable, and, finally, we argue that V cannot be computable in view of Lemma 4.

Proof. Assume the universal c.e. prefix code $V \subseteq X^*$ to be computable. We construct a sequence of finite prefix codes $(D_i)_{i\in\mathbb{N}}$ and a sequence of numbers $(l_i)_{i\in\mathbb{N}}$ such that:

1 $D_k \subset D_{k+1}$.
2 $D_k \subseteq X^{\leqslant l_k}$.
3 $D_k \cup (V \cap X^{\geqslant (l_k+1)})$ is a prefix code.
4 $|D_k \cap X^{\leqslant l_k}| > |V \cap X^{\leqslant l_k+k}|$.

We start with $v_0 := \min_{\leqslant_{\text{qlex}}} V$, that is, v_0 is the minimum of $V \subseteq X^*$ with respect to the quasi-lexicographical ordering†, and put $l_0 := |v_0| + 1$ and

$$D_0 := (V \cap X^{\leqslant l_0}) \setminus \{v_0\} \cup v_0 \cdot X\,.$$

Then it is obvious that conditions 2 and 4 are fulfilled and, since V is a prefix code, condition 3 is also fulfilled.

Next, suppose D_{i-1} has already been constructed in such a way that conditions 1 to 4 are fulfilled. We construct D_i as follows.

We let $v_i := \min_{\leqslant_{\text{qlex}}}(V \cap X^{\geqslant (l_{i-1}+1)})$ and define the number m_i as the smallest number $m \in \mathbb{N}$ such that

$$|D_{i-1} \cup v_i \cdot X^m \cup \{v \mid v \in V \setminus \{v_i\} \wedge |v_i| \leqslant |v| \leqslant |v_i| + m\}| > |V \cap X^{\leqslant (|v_i|+m+i)}|\,.$$

The number m_i exists because, in view of Lemma 6, we already have

$$|v_i \cdot X^m| = r^m > |V \cap X^{\leqslant (|v_i|+m+i)}|$$

for some $m \in \mathbb{N}$.

Observe also that the three sets D_{i-1}, $v_i \cdot X^m$ and $\{v \mid v \in V \setminus \{v_i\} \wedge |v_i| \leqslant |v| \leqslant |v_i| + m\}$ are pairwise disjoint.

Then we set $l_i := |v_i| + m_i$ and

$$D_i := D_{i-1} \cup v_i \cdot X^{m_i} \cup \{v \mid v \in V \setminus \{v_i\} \wedge |v_i| \leqslant |v| \leqslant l_i\}\,.$$

It remains to verify that D_i fulfils conditions 1 to 4. Conditions 1 and 2 are easy to see, and condition 4 follows from the definition of the number m_i. In order to verify the third property, observe that

$$D_i \cup (V \cap X^{\geqslant (l_i+1)}) = D_{i-1} \cup (V \cap X^{\geqslant (l_{i-1}+1)} \setminus \{v_i\}) \cup v_i \cdot X^{m_i}\,,$$

where $D_{i-1} \cup (V \cap X^{\geqslant (l_{i-1}+1)})$ is, by the induction hypothesis, a prefix code. Assume now that $w \sqsubset v$ for some strings $w, v \in D_i \cup (V \cap X^{\geqslant (l_i+1)})$.

The case in which both strings w, v do not belong to $v_i \cdot X^{m_i}$ is impossible by the hypothesis. For the case with $v \in v_i \cdot X^{m_i}$, we obtain $w \sqsubset v_i$ or $v_i \sqsubset w$, contradicting the

† Since V is assumed to be computable, v_0 and the subsequent v_i can be effectively computed.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

fact that $D_{i-1} \cup (V \cap X^{\geqslant(l_{i-1}+1)})$ is a prefix code. The case $w \in v_i \cdot X^{m_i}$ yields $v_i \sqsubset v$, which also contradicts the hypothesis.

Finally, it is obvious from the above construction that $D := \bigcup_{i\in\mathbb{N}} D_i$ is computable if V is computable, and according to Lemma 4, the code V cannot be universal c.e. □

3.3. *Non-maximality of c.e. prefix codes*

In Section 3.1 we have seen that a universal c.e. prefix code V is large in the sense that every c.e. prefix code can be one-to-one and computably embedded into V. In this section we are going to investigate how large universal c.e. prefix codes are if we consider set-theoretical containment rather than embeddability. To this end, we recall that a prefix code $V \subseteq X^*$ is called *maximal* provided that for every prefix code $W \subseteq X^*$, we have $V \subseteq W$ implies $W = V$.

The following result from Berstel and Perrin (1985) gives an alternative characterisation of maximal prefix codes.

Lemma 9. A code $V \subseteq X^*$ is a maximal prefix code if and only if V is a prefix code and for every $v \in X^*$ there is a $w \in V$ such that $v \sqsubseteq w$ or $w \sqsubseteq v$.

Next, we note that for c.e. prefix codes, maximality implies computability.

Lemma 10. If $V \subseteq X^*$ is a c.e. maximal prefix code, then V is computable.

Proof. In order to decide whether $v \in X^*$ belongs to V, we enumerate V as long as a string $w \in V$ with $v \sqsubseteq w$ or $w \sqsubseteq v$ appears. Then $v \in V$ if and only if $v = w$. □

With Corollary 8, we obtain the following corollary.

Corollary 11. No universal c.e. prefix code is (contained in) a maximal c.e. prefix code.

It should be noted that the property in Corollary 11 is not typical for universal c.e. prefix codes, as it can also hold for certain computable prefix codes – we give an example of a computable prefix code that is not contained in a computable maximal prefix code.

Example 12. Let $X = \{0,1\}$ and consider a set $K \subseteq \mathbb{N}$ that is infinite c.e. but not computable. Then there is a one-to-one computable function $\mathbb{N} \to K$ enumerating K. Since the graph of f is computable, the prefix code $V_K := \{0^{f(|w|)} \cdot 1 \cdot w \mid w \in \{0,1\}^*\} \subseteq \{0,1\}^*$ is also computable, but not maximal.

Assume $V_K \subseteq V$ for some computable maximal prefix code $V \subseteq \{0,1\}^*$. Observe that, since V is a prefix code and K is infinite, $0^* \cap V = \varnothing$. Thus, for every $n \in \mathbb{N}$, V contains a string of the form $0^n \cdot 1 \cdot v$.

Therefore, in order to decide whether $n \in K$, one enumerates V as long as a string of the form $0^n \cdot 1 \cdot v$ appears and then tests whether $f(|v|) = n$. □

If V is a prefix code that satisfies Kraft's inequality with equality, that is, one for which $\sum_{w\in V} r^{-|w|} = 1$, then it is maximal; the converse implication is true for finite codes, but false in general. See Berstel and Perrin (1985) for further details.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

It should be mentioned that, unlike the case of finite codes, for every function $f : \mathbb{N} \to \mathbb{N}$ with $\sum_{i\geqslant 0} r^{-f(i)} \leqslant 1$, there is a maximal prefix code $V_f = \{w_i \mid i \geqslant 0\} \subseteq X^*$ such that $|w_i| = f(i)$ (see Staiger (2007)). If f is computable and monotone, then V_f is computable also. (More precisely, if f is monotone, then V_f is computable in f.)

On the other hand, it is known from the Kraft–Chaitin Theorem (see Calude (2002), for example) that for every computable function $f : \mathbb{N} \to \mathbb{N}$ with $\sum_{i\geqslant 0} r^{-f(i)} \leqslant 1$, there is a universal c.e. prefix code $V_f = \{w_i \mid i \geqslant 0\} \subseteq X^*$ such that $|w_i| = f(i)$.

There is, however, no computable procedure assigning to a (non-monotone) computable function $f : \mathbb{N} \to \mathbb{N}$ with $\sum_{i\geqslant 0} r^{-f(i)} \leqslant 1$ a c.e. maximal prefix code $V_f = \{w_i \mid i \geqslant 0\}$ such that $|w_i| = f(i)$ for every $i \geqslant 0$.

To show this, we use the following property.

Proposition 13. If $V \subseteq X^*$ is c.e. (computable), its set of lengths $\{|w| \mid w \in V\} \subseteq \mathbb{N}$ is also c.e. (computable).

Assuming now that a computable function f_K that enumerates $\{i + 2 \mid i \in K\}$, where $K \subseteq \mathbb{N}$ is c.e. but not computable, yields, in a computable way, a maximal prefix code V_{f_K}, we can, by virtue of Lemma 10 and Proposition 13, compute K, contradicting the uncomputability of K.

4. Information-theoretic size

In the preceding section we have shown that universal c.e. prefix codes are not maximal with respect to set inclusion, so they are in some sense not large. This observation is supported by the fact mentioned in the proof of Lemma 6 that their language-theoretic density is 0.

Here we derive results on universal c.e. prefix codes that show that they are large in some information-theoretic respect. To this end, we consider a different quantity, which measures the amount of information necessary to print a string of length n in a certain language.

For a language $W \subseteq X^*$, let its *structure generating function* (cf. Kuich (1970), Staiger (1993) and Staiger (2005)) be $\mathfrak{s}_W : [0, \infty) \to [0, \infty]$ where $\mathfrak{s}_W(t) := \sum_{n\in\mathbb{N}} |W \cap X^n| \cdot t^n$. Then

$$\mathfrak{s}_W(r^{-\alpha}) = \sum\nolimits_{w\in W} r^{-\alpha\cdot|w|},$$

and $\mathfrak{s}_W(r^{-\alpha}) = \infty$ means that the function $s_W(n) := |W \cap X^n|$ cannot be upper-bounded by $r^{\alpha\cdot n}$.

4.1. *The structure generating function of a c.e. prefix code*

From Kraft's inequality, it is known that for any code $D \subseteq X^*$ and $\alpha = 1$ we have the bound $\sum_{w\in D} r^{-|w|} \leqslant 1$. If $\mathfrak{s}_D$ is a rational function, in particular, when D is a regular language, we have $\mathfrak{s}_D(\frac{1}{r} + \varepsilon) < \infty$ for some $\varepsilon > 0$. This amounts to $\sum_{w\in D} r^{-\alpha\cdot|w|} < \infty$ for some $\alpha < 1$.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

In this section we are going to show that universal c.e. prefix codes do not have this behaviour, that is, they satisfy $\mathfrak{s}_D(r^{-\alpha}) = \sum_{w\in D} r^{-\alpha\cdot|w|} = \infty$ for all α, $0 \leqslant \alpha < 1$. We will also investigate some reasons for and consequences of this behaviour.

We start with a consequence of Theorem 1: a technical result from which we derive a simplification of the proof of Theorem 3.2.(b) in Tadaki (2002).

Lemma 14. Let $D \subseteq X^*$ be a c.e. prefix code and $\alpha \in (0,\infty)$, and let U be a universal machine. If D is finite or $\alpha \geqslant 1$, then there is a constant k such that

$$\sum\nolimits_{w\in D} r^{-\alpha\cdot|w|} \leqslant r^{\alpha\cdot k}\cdot\sum\nolimits_{w\in D} r^{-\alpha\cdot H_U(w)} \leqslant r^{\alpha\cdot k}\cdot\sum\nolimits_{v\in\mathrm{dom}(U)} r^{-\alpha\cdot|v|}\,.$$

Remark. Note that for infinite D and $\alpha < 1$ the sum $\sum_{w\in D} r^{-\alpha\cdot H_U(w)}$ is always infinite. The more general fact that $\sum_{w\in W} r^{-\alpha\cdot H_U(w)}$ diverges for $\alpha < 1$ and arbitrary infinite c.e. $W \subseteq X^*$ was derived in Equation (49) of Tadaki (2002). For the sake of completeness, we prove it as Lemma 15 below.

Proof of Lemma 14. We use the one–one function φ of Corollary 2. In order to verify the first inequality, observe that the third condition of Corollary 2 implies $H_U(w) \leqslant |\varphi(w)| \leqslant |w| + k$, for $w \in D$. Now the second inequality follows immediately from the fact that $\{v \mid U(v) \in D \wedge |v| = H(U(v))\} \subseteq \mathrm{dom}(U)$. □

Lemma 15. Let W be an arbitrary infinite c.e. subset of X^* and $0 \leqslant \alpha < 1$. Then

$$\sum\nolimits_{w\in W} r^{-\alpha\cdot H_U(w)} = \infty\,.$$

Proof. Let $f : \mathbb{N} \to X^*$ be a computable one–one function enumerating W. Then every string $w \in W$ has a unique pre-image $n \in \mathbb{N}$. Hence

$$\sum\nolimits_{w\in W} r^{-\alpha\cdot H_U(w)} = \sum\nolimits_{n\in\mathbb{N}} r^{-\alpha\cdot H_U(f(n))}\,.$$

Now, $H_U(f(n)) \leqslant \log_r n + 2\cdot\log_r\log_r n + c$ for $n \geqslant r$, and if n_α, depending on α with $0 \leqslant \alpha < 1$, is large enough, we have $2\alpha\cdot\log_r\log_r n \leqslant (1-\alpha)\cdot\log_r n$ whenever $n \geqslant n_\alpha$. Thus, we obtain

$$r^{-\alpha\cdot H_U(f(n))} \geqslant c'\cdot\frac{1}{n},$$

and hence the series diverges. □

As a corollary to Lemma 15, we obtain Theorem 3.2.(b) of Tadaki (2002).

Theorem 16. For $0 \leqslant \alpha < 1$ and every universal machine U, the series $\sum_{v\in\mathrm{dom}(U)} r^{-\alpha\cdot|v|}$ diverges.

Our proof of Lemma 15 shows that every infinite c.e. subset W of X^* is enumerated starting with low complex strings. This observation is supported by Kolmogorov's result (*cf.* Zvonkin and Levin (1970, Theorem 1.3) or Staiger (1993, Theorem 2.9)) that a string w of length n in every c.e. subset $W \subseteq X^*$ has a complexity $H(w)$ bounded by $\log_r |W \cap X^n| + o(n)$.

The 'conclusion' that the complements of c.e. subsets consist of only highly complex strings is, however, not true. We will use Staiger (1993, Theorem 2.9), which proves a

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

result analogous to the above-mentioned Kolmogorov theorem for complements of c.e. subsets of X^*. We will exploit this construction to show that an analogue of Lemma 15 is also true for a large class of complements of c.e. subsets of X^*.

As usual, a language $W \subseteq X^*$ is called *sparse* if there is a polynomial $p(n)$ such that $|W \cap X^n| \leqslant p(n)$ for every $n \in \mathbb{N}$.

Theorem 17. Let $W \subseteq X^*$ be the complement of c.e. subset of X^*, and let W be non-sparse. Then, for all $0 \leqslant \alpha < 1$, we have

$$\sum_{w \in W} r^{-\alpha \cdot H_U(w)} = \infty .$$

Proof. It is shown in the proof of Staiger (1993, Theorem 2.9) that if $W \subseteq X^*$ is the complement of c.e. subset, then there is a computable partial function $\psi :\subseteq X^* \times \mathbb{N} \to X^*$ such that:

1 $|\psi(\pi, n)| = n$ whenever $(\pi, n) \in \mathrm{dom}(\psi)$.
2 For every $w \in W$ there is a π, $|\pi| \leqslant \lceil \log_r |W \cap X^{|w|}| \rceil$ such that $\psi(\pi, |w|) = w$.

This function ψ is transformed into a computable prefix (partial) function φ as follows:

$$\varphi(v) := \begin{cases} \psi(\pi, n), & \text{if } v = \overline{\mathsf{string}_r(n)} \cdot \overline{\mathsf{string}_r(|\pi|)} \cdot \pi \\ & \text{for some } (\pi, n) \in X^* \times \mathbb{N}, \text{ and} \\ \text{undefined}, & \text{otherwise.} \end{cases}$$

Clearly, our construction shows that $\mathrm{dom}(\varphi)$ is prefix-free. Moreover, for every $w \in W \cap X^n$ there is a π' with

$$|\pi'| \leqslant \log_r |W \cap X^n| + 2 \cdot \log_r \log_r |W \cap X^n| + 2 \cdot \log_r n + 6$$

such that $\varphi(\pi') = w$[†]. Since $\log_r |W \cap X^n| \leqslant n$, we get

$$H_U(w) \leqslant \log_r |W \cap X^n| + 4 \cdot \log_r n + c$$

for all $w \in W \cap X^n$ where the constant c is suitably chosen. Then

$$\sum_{w \in W \cap X^n} r^{-\alpha \cdot H_U(w)} \geqslant |W \cap X^n|^{1-\alpha} \cdot \frac{1}{n^{4\alpha}} \cdot r^{-\alpha \cdot c}$$

whenever $W \cap X^n \neq \varnothing$.

Next we use the assumption that W is non-sparse. Then, for every α, $0 \leqslant \alpha < 1$, there are infinitely many n such that $|W \cap X^n| \geqslant n^{k(\alpha)}$ where $k(\alpha) := \lceil \frac{4 \cdot \alpha}{1-\alpha} \rceil$. Thus

$$\sum_{w \in W \cap X^n} r^{-\alpha \cdot H_U(w)} \geqslant r^{-\alpha \cdot c}$$

for infinitely many $n \in \mathbb{N}$, hence the series

$$\sum_{w \in W} r^{-\alpha \cdot H_U(w)} = \sum_{n \in \mathbb{N}} \sum_{w \in W \cap X^n} r^{-\alpha \cdot H_U(w)}$$

diverges. □

[†] Here it is understood that $\log_r \alpha := 0$ for $\alpha \leqslant 1$.

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

As a corollary to Lemma 15 and Theorem 17 we obtain the following generalisation of Theorem 16.

Corollary 18. Let U be a universal prefix machine, let $W \subseteq X^*$ be computably enumerable or a non-sparse complement of a computably enumerable language and let $D = \{\pi : \pi \in \mathrm{dom}(U) \wedge U(\pi) \in W\}$. Then $\sum_{w \in D} r^{-\alpha \cdot |w|} = \infty$ for all $\alpha < 1$.

4.2. *The entropy of c.e. prefix codes*

As mentioned above, the convergence of the series $\mathfrak{s}_W(r^{-\alpha}) = \sum_{w \in W} r^{-\alpha \cdot |w|}$, for $0 \leqslant \alpha < 1$, depends on the numbers $|W \cap X^n|$. The unique value $\mathsf{H}_W \in [0, 1]$ such that $\sum_{w \in W} r^{-\alpha \cdot |w|}$ converges for all $\alpha > \mathsf{H}_W$ is known as the *entropy* of the language W. It can be calculated as follows (see Kuich (1970), Staiger (1993) and Staiger (2005)):

$$\mathsf{H}_W = \limsup_{n \to \infty} \frac{\log_r(|W \cap X^n| + 1)}{n}. \tag{2}$$

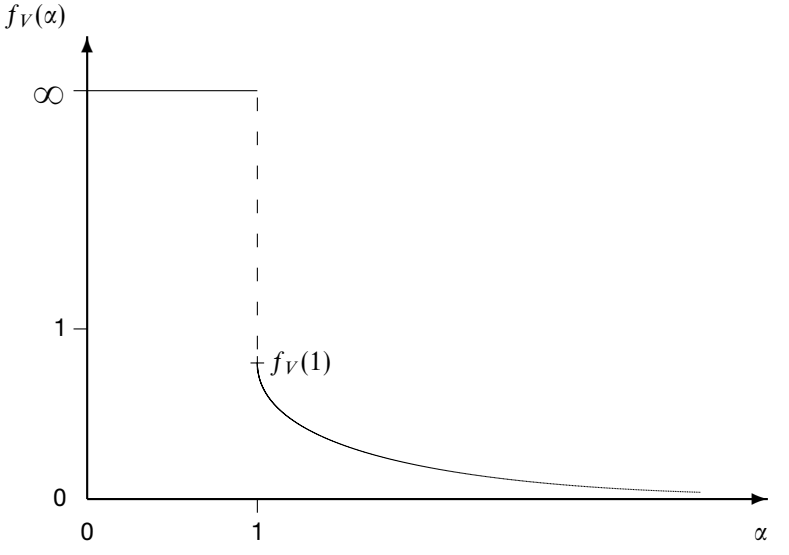
Now Corollary 16 yields the following corollary.

Corollary 19. Let $V \subseteq X^*$ be a universal c.e. prefix code. Then $\mathsf{H}_V = 1$.

Moreover, for a universal c.e. prefix code $V \subseteq X^*$, the function

$$f_V(\alpha) := \sum_{w \in V} r^{-\alpha \cdot |w|}$$

has the following typical plot:



We can substitute the upper limit in Corollary 19 by the lower one.

Theorem 20. Let $V \subseteq X^*$ be a universal c.e. prefix code. Then, the lower entropy of V is 1:

$$\liminf_{n \to \infty} \frac{1}{n} \log_r(|V \cap X^{\leqslant n}| + 1) = 1\,.$$

https://doi.org/10.1017/S0960129508007238 Published online by Cambridge University Press

Proof. Consider a universal prefix machine U such that $\mathrm{dom}(U) \subseteq V$, and consider the one-to-one mapping that maps every string $w \in X^*$ to a shortest π_w such that $U(\pi_w) = w$. It is known that $|\pi_w| \leqslant |w| + 2 \cdot \log_r |w| + c$ for some $c \in \mathbb{N}$. Consequently, $|\mathrm{dom}(U) \cap X^{\leqslant (n+2\cdot \log_r n + c)}| \geqslant r^n$, and the assertion follows. □

The property of Theorem 20 is, however, not only fulfilled by universal c.e. prefix codes. There are even languages of low complexity, more precisely, simple deterministic context-free languages (see Autebert *et al.* (1997) for a definition) that also have a lower entropy of 1. We will now give an example generalising Theorem 10 of Calude and Stay (2006) to the case $|X| > 2$.

Example 21. As in Kuich (1970) and Staiger (2005), we consider the Łukasiewicz-language $Ł_r \subseteq X^*$ defined by the equation

$$Ł_r = \{1, \ldots, r-1\} \cup 0 \cdot Ł_r^r .$$

By considering Raney sequences (*cf.* Graham *et al.* (1989)), it was shown in Kuich (1970) that for the language $W_r \subseteq \{0,1\}$ defined by the equation $W_r = 1 \cup 0 \cdot W_r^r$, we have

$$|W_r \cap \{0,1\}^{r\cdot n+1}| = \frac{(n\cdot r)!}{n! \cdot ((r-1)n+1)!} = \frac{1}{(r-1)\cdot n + 1} \binom{n\cdot r}{n},$$

and $|W_r \cap \{0,1\}^l| = 0$ if $l \not\equiv 1 \pmod r$. Moreover, every string $w \in W_r$ of length $|w| = n\cdot r+1$ has exactly n occurrences of the letter 0.

The strings of $Ł_r$ can be obtained by substituting the letter 1 by letters from $\{1, \ldots, r-1\}$. Thus

$$|Ł_r \cap X^{r\cdot n+1}| = \frac{1}{(r-1)\cdot n + 1} \binom{n\cdot r}{n} \cdot (r-1)^{n(r-1)+1} .$$

Using the inequality

$$\binom{n\cdot r}{n} > \frac{1}{\sqrt{n}} \cdot \frac{r^{r(n-1)+1}}{(r-1)^{(r-1)(n-1)}},$$

for $n \geqslant 3$ (from Stănică (2001, Corollary 2.9)), we get

$$|Ł_r \cap X^{r\cdot n+1}| \geqslant \left(\frac{r-1}{r}\right)^r \cdot \frac{1}{((r-1)\cdot n + 1)\cdot \sqrt{n}} \cdot r^{r\cdot n+1},$$

which proves that $\liminf_{l\to\infty} \frac{1}{l} \log_r |Ł_r \cap X^{\leqslant l}| = 1$.

Using Lemma 4, our Example 21 yields an alternative proof of Theorem 20.

Acknowledgments

We would like to thank A. Nies and the anonymous referee for comments that helped us improve the presentation of this paper.

References

Autebert, J.-M., Berstel, J. and Boasson, L. (1997) Context-free languages and pushdown automata. In: Rozenberg, G. and Salomaa, A. (eds.) *Handbook of Formal Languages*, Vol. 1, Springer-Verlag 111–174.

Berstel, J. and Perrin, D. (1985) *Theory of Codes*, Academic Press.

Calude, C. S. (2002) *Information and Randomness: An Algorithmic Perspective*, 2nd Edition, Revised and Extended, Springer-Verlag.

Calude, C. S. and Stay, M. A. (2006) Natural halting probabilities, partial randomness, and zeta functions. *Inform. and Comput*. **204** 1718–1739.

Chaitin, G. J. (1987) *Algorithmic Information Theory* (3rd printing 1990), Cambridge University Press.

Downey, R. and Hirschfeldt, D. (to appear) *Algorithmic Randomness and Complexity*, Springer-Verlag.

Graham, R. L., Knuth, D. E. and Patashnik, O. (1989) *Concrete Mathematics: A Foundation for Computer Science*, Addison-Wesley.

Kuich, W. (1970) On the entropy of context-free languages. *Inform. and Control* **16** 173–200.

Levenšteĭn, V. I. (1968) The redundancy and delay of decodable coding of natural numbers. *Problemy Kibernet*. **20** 173–179.

Nies, A. (2007) Personal communication.

Nies, A. (to appear) *Computability and Randomness*, Oxford University Press.

Staiger, L. (1993) Kolmogorov complexity and Hausdorff dimension. *Inform. and Comput*. **103** 159–194.

Staiger, L. (2005) The entropy of Łukasiewicz languages. *RAIRO–Theoretical Informatics and Applications* **39** (4) 621–640.

Staiger, L. (2007) On maximal prefix codes. *Bull. EATCS* **91** 205–207.

Stănică, P. (2001) Good lower and upper bounds on binomial coefficients. *J. Ineq. in Pure and Appl. Math.* **2** 1–5.

Tadaki, K. (2002) A generalization of Chaitin's halting probability Ω and halting self-similar sets. *Hokkaido Math. J*. **31** 219–253.

Zvonkin, A. K. and Levin, L. A. (1970) Complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Math. Surveys* **25** 83–124.