Fundamenta Informaticae 138 (2015) 77–84
DOI 10.3233/FI-2015-1199
IOS Press

Universality and Almost Decidability

Cristian S. Calude*
Department of Computer Science, University of Auckland
Private Bag 92019, Auckland, New Zealand
cristian@cs.auckland.ac.nz

Damien Desfontaines
École Normale Supérieure
45 rue d'Ulm, 75005 Paris, France
desfonta@clipper.ens.fr

Abstract. We present and study new definitions of universal and programmable universal unary functions and consider a new simplicity criterion: almost decidability of the halting set. A set of positive integers S is almost decidable if there exists a decidable and generic (i.e. a set of natural density one) set whose intersection with S is decidable. Every decidable set is almost decidable, but the converse implication is false. We prove the existence of infinitely many universal functions whose halting sets are generic (negligible, i.e. have density zero) and (not) almost decidable. One result—namely, the existence of infinitely many universal functions whose halting sets are generic (negligible) and not almost decidable—solves an open problem in [9]. We conclude with some open problems.

Keywords: Universal function, halting set, density, generic and negligible sets, almost decidable set

1. Universal Turing Machines and Functions

The first universal Turing machine was constructed by Turing [19, 20]. In Turing's words:

> *...a single special machine of that type can be made to do the work of all. It could in fact be made to work as a model of any other machine. The special machine may be called the universal machine.*

*Address for correspondence: Department of Computer Science, University of Auckland, Private Bag 92019, Auckland, New Zealand

Shannon [18] proved that two symbols were sufficient for constructing a universal Turing machine providing enough states can be used. According to Margenstern [12]: "Claude Shannon raised the problem of what is now called the *descriptional complexity* of Turing machines: how many states and letters are needed in order to get universal machines?" Notable universal Turing machines include the machines constructed by Minsky (7-state 4-symbol) [15], Rogozhin (4-state 6-symbol) [17], Neary–Woods (5-state 5-symbol) [16]. Herken's book [10] celebrates the first 50 years of universality. Woods and Neary presents a survey in [21]; Margenstern's paper [12, p. 30–31] presents also a time line of the main results.

Roughly speaking, a universal machine is a machine capable of simulating any other machine. There are a few definitions of universality, the most important being *universality in Turing's sense* and *programmable universality* in the sense of Algorithmic Information Theory [1, 7].

In the following we denote by $\mathbb{Z}^+$ the set of positive integers $\{1, 2, \ldots\}$, and $\overline{\mathbb{Z}^+} = \mathbb{Z}^+ \cup \{\infty\}$. The cardinality of a set S is denoted by $\#S$. The domain of a partial function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ is $\mathrm{dom}(F) = \{x \in \mathbb{Z}^+ \mid F(x) \neq \infty\}$. We assume familiarity with the basics of computability theory [6, 13].

We define now universality for unary functions.

A partially computable function $U\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ is called *(Turing) universal* if there exists a partially computable function $C_U\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ such that for any partially computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ there exists an integer $g_{U,F}$ (called a *Gödel number* of F for U) such that for all $x \in \mathbb{Z}^+$ we have: $U\left(C_U\left(g_{U,F}, x\right)\right) = F\left(x\right)$.

Following [14, 3] we say that a partially computable function $U\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ is *programmable universal* if for every partially computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ there exists a constant $k_{U,F}$ such that for every $x \in \mathbb{Z}^+$ there exists $y \leq k_{U,F} \cdot x$ with $U(y) = F(x)$.[1]

Theorem 1. A partially computable function $U\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ is programmable universal iff there exists a partially computable function $C_U\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ such that for any partially computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ there exist two integers $g_{U,F}, c_{U,F}$ such that for all $x \in \mathbb{Z}^+$ we have

$$U\left(C_U\left(g_{U,F}, x\right)\right) = F\left(x\right) \tag{1}$$

and

$$C_U\left(g_{U,F}, x\right) \leq c_{U,F} \cdot x. \tag{2}$$

Proof:
First we construct a partially computable function $V\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ and a partially computable function $C_V\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ such that for every partially computable function F, (1) and (2) are satisfied. Indeed, the classical Enumeration Theorem [6] shows the existence of a partial computable function $\Gamma\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ such that for every partial computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ there exists $e \in \mathbb{Z}^+$ such that $F(x) = \Gamma(e, x)$, for all $x \in \mathbb{Z}^+$. Consider the computable function $f\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ such that the binary expansion of $f(e, x)$ is obtained by prefixing the binary expansion of x with the binary expansion of $2e + 1$. Then α is injective because if $e_1 e_2 \ldots e_n$ and $x_1 x_2 \ldots x_m$ are the binary expansions of e and x, respectively, then $e_1 0 e_2 0 \ldots e_n 1 x_1 x_2 \ldots x_m$ is the binary expansion of $f(e, x)$ from which we can uniquely recover e and x. If $f_1, f_2\colon \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ are computable partial inverses of f, i.e.

[1] For the programming-oriented reader we note that the property "programmable universal" corresponds to being able to write a compiler.

$f(f_1(x), f_2(x)) = x$, for all $x \in f(\mathbb{Z}^+ \times \mathbb{Z}^+)$, then the function $V(x) = \Gamma(f_1(x), f_2(x))$ has (1) and (2) for $C_V = f$.[2]

Let U be programmable universal, that is, for every partially computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ there exists a constant $k_{U,F}$ such that for every $x \in \mathbb{Z}^+$ there exists $y \leq k_{U,F} \cdot x$ with $U(y) = F(x)$. We shall use V to prove that U satisfies the condition in the statement of the theorem.

Let $b\colon \mathbb{Z}^+ \times \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ be a computable bijection and b_1, b_2 the components of its inverse.

We define the partially computable function C_U as follows. We consider first the set $S(z,x) = \{y \in \mathrm{dom}(U) \mid y \leq b_1(z) \cdot x, U(y) = V(C_V(b_2(z), x))\}$ and then we define $C_U(z,x)$ to be the first element of $S(z,x)$ according to some computable enumeration of $\mathrm{dom}(U)$. Formally, let E be a computable one-one enumeration of $\mathrm{dom}(U)$ and define

$$C_U(z,x) = E\left(\inf\{y \mid E(y) \leq b_1(z) \cdot x \text{ and } U(E(y)) = V(C_V(b_2(z), x))\}\right).$$

We now prove that U satisfies the condition in the statement of the theorem via C_U. To this aim let F be a partially computable function and let $g_{V,F}, c_{V,F}$ be the constants associated to V and F.

Put $g_{U,F} = b(k_{U,F}, g_{V,F})$ and $c_{U,F} = k_{U,F}$.

We have:

$$\begin{aligned} C_U(g_{U,F}, x) &= E\left(\inf\{y \mid E(y) \leq b_1(g_{U,F}) \cdot x \text{ and } U(E(y)) = V(C_V(b_2(g_{U,F}), x))\}\right) \\ &= E\left(\inf\{y \mid E(y) \leq k_{U,F} \cdot x \text{ and } U(E(y)) = V(C_V(g_{V,F}, x))\}\right) \\ &= E\left(\inf\{y \mid E(y) \leq k_{U,F} \cdot x \text{ and } U(E(y)) = F(x)\}\right) \\ &\leq k_{U,F} \cdot x = c_{U,F} \cdot x, \end{aligned}$$

and $U(C_U(g_{U,F}, x)) = F(x)$.

Conversely, if V satisfies (1) and (2) with the partially computable function C_V, then V is programmable universal: given a partially computable function F and $x \in \mathbb{Z}^+$, $y = C_V(g_{V,F}, x)$ and $k_{V,F} = c_{V,F}$. □

Universal and programmable universal functions exist and can be effectively constructed. Every programmable universal function is universal, but the converse implication is false.

2. The Halting Set and Almost Decidability

Interesting classes of Turing machines have decidable halting sets: for example, Turing machines with two letters and two states [12]. In contrast, the most (in)famous result in computability theory is that *the halting set Halt*$(U) = \mathrm{dom}(U)$ *of a universal function U is undecidable.*

However, the halting set Halt(U) is computably enumerable (see [6, 13]). How "undecidable" is Halt(U)? To answer this question we formalise the following notion: a set S is "almost decidable" if there exists a "large" decidable set whose intersection with S is also decidable. In other words, the undecidability of S can be located to a "small" set.

[2]This construction suggests that the function C_U in the definition of a universal function may be taken to be computable.

To define "large" sets we can employ measure theoretical or topological tools adapted to the set of positive integers (see [1]). In what follows we will work with the *(natural) density* on $\mathcal{P}(\mathbb{Z}^+)$. Its motivation is the following. If a positive integer is "randomly" selected from the set $\{1, 2, \ldots, N\}$, then the probability that it belongs to a given set $A \subset \mathbb{Z}^+$ is

$$p_N(A) = \frac{\#(\{1, \ldots, N\} \cap A)}{N}.$$

If $\lim_{N \longrightarrow \infty} p_N(A)$ exists, then the set $A \subset \mathbb{Z}^+$ has *density*:

$$d(A) = \lim_{N \longrightarrow \infty} \frac{\#\{1 \leq x \leq N \mid x \in A\}}{N}.$$

Definition 2. A set is *generic* if it has density one; a set of density zero is called *negligible*. A set $S \subset \mathbb{Z}^+$ is *almost decidable* if there exists a generic decidable set $R \subset \mathbb{Z}^+$ such that $R \cap S$ is decidable.

Every decidable set is almost decidable, but, as we shall see below, there exist almost decidable sets which are not decidable. A set which is not almost decidable contains no generic decidable subset; of course, this result is non-trivial if the set itself is generic.

Theorem 3. ([9], Theorem 1.1)
There exists a universal Turing machine whose halting set is negligible and almost decidable (in polynomial time).

A single semi-infinite tape, single halt state, binary alphabet universal Turing machine satisfies Theorem 3; other examples are provided in [9].

Negligibility reduces to some extent the power of almost decidability in Theorem 3. This deficiency is overcome in the next result: the price paid is in the redundancy of the universal function.

Proposition 4. There exist infinitely many universal functions whose halting sets are generic and almost decidable (in polynomial time).

Proof:
Let V be a universal function and define U by the formula:

$$U(x) = \begin{cases} V(y), & \text{if } x = y^2, \text{ for some } y \in \mathbb{Z}^+, \\ 0, & \text{otherwise}. \end{cases}$$

Clearly, U is universal, Halt(U) is generic, the set $S = \{y \in \mathbb{Z}^+ \mid y \neq x^2 \text{ for every } x \in \mathbb{Z}^+\}$ is generic and decidable (in polynomial time) and $S \cap$ Halt(U) is generic and decidable (in polynomial time). □

Corollary 5. There exist infinitely many almost decidable but not decidable sets.

Does there exist a universal function U whose halting set is not almost decidable? This problem was left open in [9]: here we answer it in the affirmative.

Theorem 6. There exist infinitely many universal functions whose halting sets are not negligible and not almost decidable.

Proof:
We start with an arbitrary universal function V and construct a new universal function U whose halting set Halt(U) is not almost decidable.

First we define the computable function $\varphi\colon \mathbb{Z}^+ \longrightarrow \mathbb{Z}^+$ by $\varphi(n) = \max\{k \in \mathbb{Z}^+ \mid 2^{k-1} \text{ divides } n\}$.

The function φ has the following properties:

(a) $\varphi(2^{m-1}(2k+1)) = m$, for every $m, k \in \mathbb{Z}^+$, so φ outputs every positive integer infinitely many times.

(b) $\varphi^{-1}(n) = \{k \in \mathbb{Z}^+ \mid 2^{n-1} \text{ divides } k \text{ but } 2^n \text{ does not divide } k\}$.

(c) $d(\varphi^{-1}(n)) = 2^{-n}$, for all $n \in \mathbb{Z}^+$.

(d) If $S \subseteq \mathbb{Z}^+$ and $d(S) = 1$, then for every $n \in \mathbb{Z}^+$, $\varphi^{-1}(n) \cap S \neq \emptyset$.

For (d) we note that if $\varphi^{-1}(n) \cap S = \emptyset$, then $d(S) \leq 1 - 2^{-n}$, a contradiction.

Next we define $U(x) = V(\varphi(x))$ and prove that U is universal. We consider the partially computable function $C_U(z,x) = \inf\{s \in \mathbb{Z}^+ \mid \varphi(s) = C_V(z,x)\}$ and note that: 1) by (a), $\mathrm{dom}(C_U) = \mathrm{dom}(C_V)$, and 2) $\varphi(C_U(z,x)) = C_V(z,x)$, for all $(z,x) \in \mathrm{dom}(C_V)$. Consequently, for every partially computable function $F\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ we have $F(x) = V(C_V(g_{V,F}, x)) = V(\varphi(C_U(g_{V,F}, x)))$, so $g_{U,F} = g_{V,F}$.

Let us assume by absurdity that there exists a generic decidable set $S \subseteq \mathbb{Z}^+$ such that $S \cap$ Halt(U) is decidable.

Define the partial function $\theta\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ by $\theta(n) = \inf\{k \in S \mid \varphi(k) = n\}$.

As S is decidable, θ is partially computable; by (a) (φ is surjective) and by (d) (as $d(S) = 1$, for all $n \in \mathbb{Z}^+$, $\varphi^{-1}(n) \cap S \neq \emptyset$) it follows that θ is computable. Furthermore, the computable function θ has the following two properties: for all $n \in \mathbb{Z}^+$, $\varphi(\theta(n)) = n$ and $\theta(n) \in S$.

We next prove that for all $n \in \mathbb{Z}^+$,

$$n \in \text{Halt}(V) \text{ iff } \theta(n) \in S \cap \text{Halt}(U). \tag{3}$$

Indeed,

$$\begin{aligned} n \in \text{Halt}(V) &\iff V(n) < \infty \\ &\iff V(\varphi(\theta(n))) < \infty && (\varphi(\theta(n)) = n) \\ &\iff U(\theta(n)) < \infty && (\text{definition of } U) \\ &\iff \theta(n) \in \text{Halt}(U) \\ &\iff \theta(n) \in S \cap \text{Halt}(U). && (\theta(n) \in S) \end{aligned}$$

From (3) it follows that Halt(V) is decidable because $S \cap$ Halt(U) is decidable, a contradiction.

Finally, $d(\text{Halt}(U)) > 0$ because Halt$(U) = \varphi^{-1}(\text{Halt}(V))$.
By varying the universal function V we get infinitely many examples of universal functions U. □

Corollary 7. There exist infinitely many universal functions U such that for any generic computably enumerable set $S \subseteq \mathbb{Z}^+$, $S \cap \text{Halt}\,(U)$ is not decidable.

Proof:
Assume S is computable enumerable and $d(S) = 1$. If replace the computable function θ with the computable function $\Gamma(n) = E(\min\{k \in \mathbb{Z}^+ \mid \varphi(E(i)) = n\})$, where $E\colon \mathbb{Z}^+ \longrightarrow \overline{\mathbb{Z}^+}$ is a computable injective function such that $E(\mathbb{Z}^+) = S$ (S is infinite) in the proof of Theorem 6, then we prove that $S \cap \text{Halt}\,(U)$ is not decidable. □

There are six possible relations between the notions of negligible, generic and almost decidable sets. The above results looked at three of them: here we show that the remaining three possibilities can be realised too. First, it is clear that there exist non-negligible and decidable sets, hence non-negligible and almost decidable sets.

The next result is a stronger form of Theorem 6: its proof depends on a set A and works for other interesting sets as well.

Theorem 8. There exist infinitely many universal functions whose halting sets are generic and not almost decidable.

Proof:
We use a computably enumerable generic set A which has no generic decidable subset (see Theorem 2.22 in [11]) to construct a universal function as in the statement above.

Assume $A = \text{Halt}(F)$ for some partially computable function F. Let V be an arbitrary universal function and define U by:

$$U(x) = \begin{cases} V(y), & \text{if } x = y^2, \text{ for some } y \in \mathbb{Z}^+, \\ F(x), & \text{otherwise}\,. \end{cases}$$

Clearly $\text{Halt}(U)$ is universal and generic.

For the sake of a contradiction assume that $\text{Halt}(U)$ is almost decidable by S, i.e. S is a generic decidable set such that $\text{Halt}(U) \cap S$ is decidable.

We now prove that $\text{Halt}(F)$ is almost decidable by $S' = S \cap \overline{P}$, where P is the set of square positive integers (note that P is decidable and negligible) and $\overline{P}$ is the complement of P. It is clear that S' is generic and decidable, so we need only to show that Halt(F) $\cap\, S' =$ Halt(F) $\cap\, S \cap \overline{P}$ is decidable.

We note that $\text{Halt}(U)$ is a disjoint union of the sets $\{x \in \mathbb{Z}^+ \mid x = y^2, \text{ for some } y \in \text{Halt}(V)\}$ and $\text{Halt}(F) \cap \overline{P}$, and the first set is a subset of P. To test whether x is in Halt(F) $\cap\, S'$ we proceed as follows: a) if $x \in P$, then $x \notin \text{Halt}(F) \cap S'$, b) if $x \notin P$, then $x \in$ Halt(F) $\cap\, S'$ iff $x \in \text{Halt}(U) \cap S$. Hence, Halt(F) $\cap\, S'$ is decidable because $\text{Halt}(U) \cap S$ is decidable, so $\text{Halt}(U)$ is almost decidable.

We have obtained a contradiction because Halt(F) $\cap\, S'$ is a generic decidable subset of A, hence $\text{Halt}(U)$ is not almost decidable. □

Let $r \in (0, 1]$. We say that a set $S \subset \mathbb{Z}^+$ is r-*decidable* if there exists a decidable set $R \subset \mathbb{Z}^+$ such that $d(R) = r$ and $R \cap S$ is decidable; a set $S \subset \mathbb{Z}^+$ is *weakly decidable* if S is r-decidable for some $r \in (0, 1)$. With this terminology, generic sets coincide with 1-decidable sets.

Theorem 3.18 of [8] states that there is a computably enumerable generic set that has no decidable subset of density in $(0, 1)$. Using this set in the proof of Theorem 8 we get the following stronger result:

Theorem 9. There exist infinitely many universal functions whose halting sets are generic and not weakly decidable.

A simple set is a co-infinite computably enumerable set whose complement has no decidable subset; the existence of a negligible simple set is shown in the proof of Proposition 2.15 in [11]. If in the proof of Theorem 8 we use a negligible simple set instead of the computably enumerable generic set which has no generic decidable subset we obtain the following result:

Theorem 10. There exist infinitely many universal functions whose halting sets are negligible and not almost decidable.

3. A Simplicity Criterion for Universal Functions and Open Problems

Universality is one of the most important concepts in computability theory. However, not all universal machines are made equal. The most popular criterion for distinguishing between universal Turing machines is the number of states/symbols. Other three other criteria of simplicity for universal prefix-free Turing machines have been studied in [2]. The property of almost decidability is another criterion of simplicity for universal functions.

The universal function U constructed in the proof of Theorem 6 is *not* programmable universal. Theorems 2 and 8 in [4] show that the halting sets of programmable universal string functions (plain or prefix-free) are never negligible. *Are there programmable universal functions not almost decidable?*

The notion of almost decidability suggests the possibility of an approximate (probabilistic) solution for the halting problem (see also [5, 3]). Assume that the halting set is $\mathrm{Halt}(U)$ is almost decidable via the generic decidable set S and we wish to test whether an arbitrary $x \in \mathbb{Z}^+$ is in $\mathrm{Halt}(U)$. If $x \in S$, then $x \in \mathrm{Halt}(U)$ iff $x \in S \cap \mathrm{Halt}(U)$. If $x \notin S$, then we don't know whether $x \in \mathrm{Halt}(U)$ or $x \notin \mathrm{Halt}(U)$ (the undecidability is located in $\overline{S} \cap \mathrm{Halt}(U)$). Should we conclude that $x \in \mathrm{Halt}(U)$ or $x \notin \mathrm{Halt}(U)$? Density does not help because $d(\overline{S} \cap \mathrm{Halt}(U)) = d(\overline{S} \cap \overline{\mathrm{Halt}(U)}) = 0$. It is an open problem to find a solution.

The notion of almost decidability can be refined in different ways, e.g. by looking at the computational complexity of the decidable sets appearing in Theorem 6. Also, it will be interesting to study the property of *almost decidability* topologically or for other densities.

Acknowledgement

The authors have been supported in part by the Quantum Computing Research Initiatives at Lockheed Martin. We thank Laurent Bienvenu, Ludwig Staiger, Mike Stay and Declan Thompson for interesting discussions on the topic of this paper. We also thank the anonymous referees for comments which improved the paper.

References

[1] C. S. Calude. *Information and Randomness. An Algorithmic Perspective*, 2nd Edition, Revised and Extended, Springer Verlag, Berlin, 2002.

[2] C. S. Calude. Simplicity via provability for universal prefix-free Turing machines, *Theoretical Comput. Sci.* 412 (2010), 178–182.

[3] C. S. Calude, D. Desfontaines. Anytime Algorithms for Non-Ending Computations, *CDMTCS Research Report* 463, 2014.

[4] C. S. Calude, A. Nies, L. Staiger, F. Stephan. Universal recursively enumerable sets of strings, *Theoretical Comput. Sci.* 412 (2011), 2253–2261.

[5] C. S. Calude, M. A. Stay. Most programs stop quickly or never halt, *Advances in Applied Mathematics*, 40 (2008), 295–308.

[6] S. B. Cooper. *Computability Theory*, Chapman & Hall/CRC London, 2004.

[7] R. Downey, D. Hirschfeldt. *Algorithmic Randomness and Complexity*, Springer, Heidelberg, 2010.

[8] R. G. Downey, C. G. Jockusch Jr., P. E. Schupp. `http://www.newton.ac.uk/preprints/NI12039.pdf` Asymptotic Density and Computably Enumerable Sets, Newton Institute Preprint NI12039, 2012.

[9] J. D. Hamkins, A. Miasnikov. The halting problem is decidable on a set of asymptotic probability one, *Notre Dame J. Formal Logic* 47 (4) (2006), 515–524.

[10] R. Herken. *The Universal Turing Machine: A Half-Century Survey*, Oxford University Press, Oxford, 1992.

[11] C. G. Jockusch, Jr., P. Schupp. Generic computability, Turing degrees, and asymptotic density, *Journal of the London Mathematical Society* 85 (2) (2012), 472–490.

[12] M. Margenstern. Turing machines with two letters and two states, *Complex Systems* 19 (2010), 29–43.

[13] Yu. I. Manin. *A Course in Mathematical Logic for Mathematicians*, Springer, Berlin, 1977; second edition, 2010.

[14] Yu. I. Manin. Renormalisation and computation II: time cut-off and the Halting Problem, *Math. Struct. in Comp. Science* 22 (2012), 729–751.

[15] M. Minsky. Size and structure of universal Turing machines using Tag systems, in *Recursive Function Theory, Proc. Symp. Pure Mathematics*, AMS, Providence RI, 5, 1962, 229–238.

[16] N. Neary. D. Woods. The complexity of small universal Turing machines, in S. B. Cooper, B. Loewe, A. Sorbi (eds.). *Computability in Europe 2007*, LNCS 4497, CIE, Springer, 2007, 791–798.

[17] Y. Rogozhin. A universal Turing machine with 22 states and 2 symbols, *Romanian Journal of Information Science and Technology* 1 (3) (1998), 259–265.

[18] C. Shannon. A universal Turing machine with two internal states, in *Automata Studies, Princeton*, Princeton University Press, NJ, 1956, 157–165.

[19] A. Turing. On computable numbers, with an application to the Entscheidungsproblem, *Proceedings of the London Mathematical Society* 42 (2) (1936), 230–265.

[20] A. Turing. On computable numbers, with an application to the Entscheidungsproblem: A correction, *Proceedings of the London Mathematical Society* 2, 43 (1937), 544–546.

[21] D. Woods, N. Neary. The complexity of small universal Turing machines: A survey, *Theor. Comput. Sci.* 410(4-5), (2009), 443–450.