Available online at www.sciencedirect.com
ScienceDirect
Theoretical Computer Science 377 (2007) 268–270
Theoretical Computer Science
www.elsevier.com/locate/tcs

Note

On the strongly generic undecidability of the Halting Problem

Alexander Rybalov*

Department of Mathematical Logic, Omsk State University, prospekt Mira 55a, Omsk 644077, Russia

Received 17 May 2006; received in revised form 27 January 2007; accepted 4 February 2007

Communicated by Z. Esik

Abstract

It has been shown in [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] that the classical Halting Problem for Turing machines with one-way tape is decidable on a "large" set of Turing machines (a so-called generic set). However, here we prove that the Halting Problem remains undecidable when restricted to an arbitrary "very large" set of Turing machines (a so-called strongly generic set). Our proof is independent of a Turing machine model.

Keywords: Generic complexity; Halting Problem

1. Introduction

In [1,3,4] it was shown that many classical algorithmically undecidable problems in algebra are generically decidable (even quickly decidable). This means that for every such problem there is a correct partial decision algorithm whose halting set is generic (see definition below).

This generic complexity approach has numerous applications, especially in cryptography, where cryptosystems must be based on problems which are hard in almost all (random) instances. Usually the behavior of such problems is studied in terms of average-case complexity, but generic-case complexity and generic computability are much more convenient to use in applications, in particular they allow the study of the complexity of undecidable problems.

It has been shown in [2] that the classical Halting Problem is generically decidable (even quickly decidable) for Turing machines with one-way infinite tape. Namely, the set of all Turing machines, such that the head of the machine falls off the tape (starting on the empty tape) before repeating a state, is generic. This result is model sensitive and it is an open problem to transfer it to Turing machines with two-way tape.

In this paper we prove that any generic set on which the Halting Problem is decidable cannot be very large (i.e. strongly generic). Our proof does not depend on the model of Turing machines and holds for machines with one-way and two-way tape.

The Turing machine M has a finite set of states $Q = \{q_1, \dots, q_n\}$, with q_1 designated as the start state, and an extra special state q_f designated as the final state. The head of the machine M can read and write symbols from the

* Tel.: +7 3812 670994.
E-mail addresses: rybalov@omskreg.ru, alexander.rybalov@gmail.com.

0304-3975/$ - see front matter
doi:10.1016/j.tcs.2007.02.010

alphabet $\Sigma = \{0, 1, \Box\}$ ($\Box$ is the blank symbol) while moving on a infinite tape divided into cells. The program P_M of a Turing machine M consists of a finite set of instructions (one for every $q_i \in Q$ and $s \in \Sigma$)

$$(q_i, s) \to (q_j, t, D),$$

where $q_j \in Q \cup \{q_f\}$, $t \in \Sigma$, and $D \in \{R, L\}$ — shift to the right or to the left. This instruction says that M being in state q_i with the head, reading symbol s on the tape, must change state on q_j, write symbol t instead of s and move the head (to right or left adjacent cell). We identify machine M with its program P_M.

Note that our result does not depend on the size of alphabet Σ and holds for any alphabet with at least two symbols.

At the beginning of the computation M is in the start state, an input string of 1s and 0s is written on the tape (all other cells contain the blank symbol $\Box$), and the head of M is over the cell with first symbol of the input string. The computation proceeds by iteratively performing the instructions, halting when the state q_f is reached.

Let δ be some effective coding of all Turing machines by binary strings. So a set S of programs is decidable if and only if the set $\delta(S)$ is decidable. The Halting Problem is the set *HP* of Turing machine programs P that halt when computing on input $\delta(P)$.

Following [2] we define generic sets of Turing machine programs via asymptotic density. Let $\mathcal{P}$ be the set of all programs and $\mathcal{P}_n$ be the set of all programs with precisely n non-final states. The asymptotic density of a subset $S \subseteq \mathcal{P}$ is the following limit (if it exists)

$$\rho(S) = \lim_{n\to\infty} \frac{|S \cap \mathcal{P}_n|}{|\mathcal{P}_n|}.$$

S is called generic (negligible) if $\rho(S) = 1$ ($\rho(S) = 0$). Clearly, S is generic if and only if its complement in $\mathcal{P}$ is negligible. Following [3] we call a set S strongly negligible if the sequence of portions in the limit exponentially fast converges to 0, i.e. there are constants $0 < \sigma < 1$ and $C > 0$ such that for every $n \geq 1$

$$\frac{|S \cap P_n|}{|P_n|} < C\sigma^n.$$

Similarly, S is called strongly generic if its complement is strongly negligible.

Recall that *HP* is decidable on a strongly generic set S if there is a partial computable function $f : \{1\}^* \to \{0, 1\}$ such that $S \subseteq \mathrm{Dom}(f)$ and if $f(\delta(M)) = 1$ then M halts on $\delta(M)$, and if $f(\delta(M)) = 0$ then M does not halt on $\delta(M)$, and also $f(x)$ is undefined if $x \neq \delta(M)$ for any Turing machine M. In this case, we say that f is a strongly generic decision function for *HP*. In particular, it follows that the domain $\mathrm{Dom}(f)$ is a strongly generic recursively enumerable set on which *HP* is decidable.

2. Main result

At the beginning we count the number of all programs of size n.

Lemma 1. *The number of Turing machines with n non-final states is*

$$|\mathcal{P}_n| = (6(n+1))^{3n}.$$

Proof. A program of a Turing machine with n non-final states has the following form

$$\begin{aligned}
&(q_1, 0) \to (q_{j_1}, t_1, D_1),\\
&(q_1, 1) \to (q_{j_2}, t_2, D_2),\\
&(q_1, \Box) \to (q_{j_3}, t_3, D_3),\\
&(q_2, 0) \to (q_{j_4}, t_4, D_4),\\
&(q_2, 1) \to (q_{j_5}, t_5, D_5),\\
&\dots\\
&(q_n, \Box) \to (q_{j_{3n}}, t_{3n}, D_{3n}).
\end{aligned}$$

For every instruction there are precisely 3 possibilities to chose a symbol t_i, 2 possibilities to chose a shift D_i, and $n+1$ possibilities for the next state q_{j_i}. So we have $(4(n+1))^{3n}$ possible programs. $\Box$

Denote by $C(f)$ for a partial computable function $f : \{0, 1\}^* \to \{0, 1\}$ the set of all Turing machines computing f.

Lemma 2. *For any partial computable function f the set $C(f)$ is not strongly negligible.*

Proof. Let M be a Turing machine with k non-final states computing f. For any $n > k$, one can construct a new machine M^* with n non-final states with the following program

$$\text{3}k \text{ fixed instructions of } M \begin{cases} (q_1, 0) \to \dots, \\ \dots \\ (q_k, \Box) \to \dots, \end{cases}$$

$$\text{arbitrary } 3(n-k) \text{ instructions} \begin{cases} (q_{k+1}, 0) \to \dots, \\ \dots \\ (q_n, \Box) \to \dots. \end{cases}$$

It is easy to see that M^* computes the same function as M because the new states are not attainable from the states of M, and new instructions do not affect the computation. The number of such new machines is $(6(n+1))^{3(n-k)}$. Now we have

$$a_n = \frac{|C(f) \cap P_n|}{|P_n|} > \frac{(6(n+1))^{3(n-k)}}{(6(n+1))^{3n}} = \frac{1}{(6(n+1))^{3k}}.$$

It means that the sequence a_n does not converge to 0 exponentially fast, so $C(f)$ is not strongly negligible. □

Theorem 3. *There is no strongly generic set on which the Halting Problem is decidable.*

Proof. We follow here the classical proof of undecidability of the Halting Problem. Suppose, to the contrary, that there exists a strongly generic set S on which HP is decidable. Then there exists a strongly generic decision function $f : \{0, 1\}^* \to \{0, 1\}$ for HP such that $S \subseteq \mathrm{Dom}(f)$. We may assume from the beginning that $S = \mathrm{Dom}(f)$. It is easy to see then that the following function is partial computable

$$h(x) = \begin{cases} \text{undefined}, & \text{if } f(x) = 1, \\ \text{undefined}, & \text{if } f(x) \text{ is undefined}, \\ 1, & \text{if } f(x) = 1. \end{cases}$$

Notice, that the set $\mathcal{P} \setminus S$ is strongly negligible, so by Lemma 2 the set $C(h)$ is not a subset of $\mathcal{P} \setminus S$, hence, there is a machine M from S computing h. Now let's look at the result of computation of M on the input $\delta(M)$. Observe, first, that $f(\delta(M))$ is defined since $M \in S$. If M halts on $\delta(M)$ then $f(\delta(M)) = 1$, hence $h(\delta(M))$ is undefined, so M does not halt on $\delta(M)$ — contradiction. If M does not halt on $\delta(M)$ then $f(\delta(M)) = 0$, so $h(\delta(M)) = 1$, which implies that M halts on $\delta(M)$ — contradiction. This shows that such S does not exist, as required. □

Acknowledgements

The author thanks Alexei G. Miasnikov and anonymous referees for useful remarks and comments. The author was supported by the Russian Foundation for Basic Research, grant 07-01-00392.

References

[1] A.V. Borovik, A.G. Myasnikov, V.N. Remeslennikov, Multiplicative measures on free groups, Internat. J. Algebra Comput. 13 (6) (2003) 705–731.

[2] 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.

[3] I. Kapovich, A. Myasnikov, P. Schupp, V. Shpilrain, Generic-case complexity, decision problems in group theory and random walks, J. Algebra 264 (2) (2003) 665–694.

[4] I. Kapovich, A. Myasnikov, P. Schupp, V. Shpilrain, Average-case complexity and decision problems in group theory, Adv. Math. 190 (2) (2005) 343–359.