Notre Dame Journal of Formal Logic
Volume 47, Number 4, 2006

The Halting Problem Is Decidable on a Set of Asymptotic Probability One

Joel David Hamkins and Alexei Miasnikov

Abstract The halting problem for Turing machines is decidable on a set of asymptotic probability one. The proof is sensitive to the particular computational models.

1 Introduction

The classical halting problem for Turing machines is perhaps the canonical undecidable set. Nevertheless, we provide an algorithm deciding almost all instances of it with respect to what seems to be the most natural measure, the asymptotic density of Turing machine programs. We take this result as a warning that the central concerns of computability and complexity theory do not interact well with measure, and the important issues can concentrate on sets of measure zero.

The main result places the halting problem within the "black hole" phenomenon of complexity theory, occurring when the difficulty of an unfeasible problem is confined to a very small region, a black hole, outside of which it is easily solved. For example, we should not base an encryption scheme on such a problem if the intended instances of it will respect the measure, for we should not allow that a criminal could rob the bank 95% of the time or, indeed, any significant amount of the time. The second author previously found the black hole phenomenon to arise in several group theoretic decision problems and inquired whether the halting problem itself exhibited such a black hole. Our main theorem shows that indeed it does.

The most natural method for measuring sets of Turing machine programs seems to be *asymptotic density*. The asymptotic density or probability of a set B of Turing machine programs is the limit of the proportion of all n-state programs in B as n increases. That is, if P_n is the set of all n-state programs, then the asymptotic

Received March 20, 2006; accepted July 3, 2006; printed December 28, 2006
2000 Mathematics Subject Classification: Primary, 03D10, 68Q17
Keywords: Turing machines, halting problem, decidability

probability of B is

$$\mu(B) = \lim_{n\to\infty} \frac{|B \cap P_n|}{|P_n|},$$

provided that this limit exists. If B has asymptotic probability one, for example, then for sufficiently large n, more than 99% of all n-state programs are in B, and so on, as close to 100% as desired.

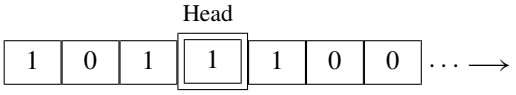
Theorem 1.1 (Main Theorem) *There is a set B of Turing machine programs such that*

(1) *B has asymptotic probability one,*
(2) *B is polynomial time decidable,*
(3) *the halting problem $H \cap B$ is polynomial time decidable.*

We will prove this theorem for a variety of computational models, but not for all computational models, while discussing the relevant details of the models in full.

2 Proof of the Main Theorem for the Standard Model

Let us begin by proving the main theorem for what we call here the *standard model* of Turing machine computation. In the standard model, we have a finite program directing the operation of a head reading and writing 0s and 1s while moving on a semi-infinite tape (infinite in one direction only), with a single halt state.



In the standard model, the Turing machine has n states $Q = \{q_1, \ldots, q_n\}$, with q_1 designated as the *start* state, plus a separate designated *halt* state, which is not counted as one of the n states. A Turing machine *program* is a function

$$p : Q \times \{0, 1\} \to (Q \cup \{halt\}) \times \{0, 1\} \times \{L, R\}.$$

The transition $p(q, i) = \langle r, j, R\rangle$, for example, directs that when the head is in state q reading symbol i, it should change to state r, write symbol j, and move one cell to the right. The computation of a program proceeds by iteratively performing the instructions of such transition rules, halting when (and if) the *halt* state is reached. If the machine attempts to move left from the left-most cell, then the head falls off the tape and computation ceases. Since the domain of the program has size $2n$ and the target space has size $4(n + 1)$, we can easily count the number of programs.

Lemma 2.1 *In the standard model, there are $(4(n + 1))^{2n}$ many n-state Turing machine programs.*

As a warm-up exercise, let us calculate the asymptotic probability of the set of programs having no transition reaching the *halt* state. Such a criterion is clearly linear time decidable (for any reasonable representation of programs by finite binary sequences), and no computation by such a program can ever reach the *halt* state.

Lemma 2.2 *In the standard model, the collection of programs having no transition reaching the* halt *state has asymptotic probability* $1/e^2$*, which is about* 13.5%.

Proof If p has no transition reaching the *halt* state, then $p : Q \times \{0,1\} \to Q \times \{0,1\} \times \{L,R\}$. Since this target set has size $4n$, the total number of such functions is $(4n)^{2n}$. The asymptotic proportion of all n-state programs with this property is therefore

$$\lim_{n\to\infty} \frac{(4n)^{2n}}{(4(n+1))^{2n}} = \lim_{n\to\infty} \left(\frac{n}{n+1}\right)^{2n} = \lim_{n\to\infty} \left[(1-\frac{1}{n+1})^n\right]^2 = 1/e^2.$$

Therefore, the asymptotic probability that a Turing machine program does not engage the *halt* state is $1/e^2$. □

Definition 2.3 The *halting problem* is the set H of programs p that halt when computing on input 0 on a tape initially filled with 0s.

For the purposes of defining the halting problem H, one should specify whether it officially counts as halting or not if the head should happen to fall off the left edge of the tape. Although the truth of the main theorem will not depend on these details, provided we adopt a uniform answer, let us be definite and regard such computations as having not officially halted, as the *halt* state was not reached. Thus, we regard H as the set of programs that eventually reach the *halt* state from an initially empty tape. To be even more specific, if the head happens to fall off the tape while executing the transition $p(q,i) = \langle r, j, L\rangle$, then we do not regard the state r as having been achieved, since this step was not completed.

Proof of the Main Theorem Using the standard model, let B be the set of programs that on input 0 either halt before repeating a state or fall off the tape before repeating a state. Clearly, B is polynomial time decidable, since we need only run a program p for at most n steps, where n is the number of states in p, to determine whether or not it is in B. It is equally clear that the halting problem is polynomial time decidable for programs p in B, since again we need only simulate p for n steps to know whether it halted or fell off. What remains is to prove that this behavior occurs with asymptotic probability one.

Lemma 2.4 *In the standard model, for any fixed input and fixed integer* $k \geq 0$*, the set of programs not repeating states within the first* k *steps has asymptotic probability one.*

Proof Just to be clear, we count a computation that halts or falls off the tape as satisfying the property if it does so before repeating a state. We calculate for large n the proportion of all n-state programs having this property, by induction on k. When $k = 0$, then all programs have the property. Suppose that the set B_k of programs having the desired property for k has asymptotic probability one, and consider B_{k+1}. Fix any ϵ, and choose n large enough so that B_k has proportion more than $1-\epsilon/2$ of all n-state programs. Among all n-state programs p in B_k, consider the probability that p is in B_{k+1}. If p leads to a computation where the head has already fallen off the tape, then of course $p \in B_{k+1}$. Otherwise, the first k steps of computation by p have led to the successive states $q_{i_0}, q_{i_1}, \ldots, q_{i_k}$, which have not yet repeated. The $(k+1)$th step of computation involves a transition rule $p(q_{i_k}, j_k) = (q_{i_{k+1}}, j_{k+1}, m_k)$, giving respectively the new state, the new bit to write on the tape and the direction to move

the head. In order for p to be in B_{k+1}, it suffices that $q_{i_{k+1}}$ must not be one of the previously used states $\{ q_{i_0}, q_{i_1}, \ldots, q_{i_k} \}$. Since there are $(n+1)-(k+1) = n-k$ many other equally likely states to choose from, the proportion of all n-state programs agreeing with p on the first k steps and satisfying the additional requirement is $\frac{n-k}{n}$. The proportion of all n-state programs in B_{k+1}, consequently, is at least $(1-\epsilon/2)(\frac{n-k}{n})$. Since $\frac{n-k}{n}$ goes to 1 as n becomes large, we may choose n large enough so that this proportion is at least $1-\epsilon$. Thus, B_{k+1} has asymptotic probability one, as desired. □

Proceeding with the main argument, let B_k be the set of programs that do not repeat a state within their first k steps of computation. The key idea is that for the first k steps of computation, the programs in B_k behave statistically like a random walk with uniform probability of going left or right. The reason is that if a program lands in a totally new state q, reading some symbol i, then among the programs landing in that situation and agreeing with the computation so far, exactly half of them will opt to move left and half will move right, precisely because nothing about state q has yet been determined. Because of this, we may make use of Polya's classical result on random walks, which we mention without proof.

Lemma 2.5 (Polya [2], see also, e.g., [1]) *In the random walk with equal likelihood of moving left or right on a semi-infinite tape, beginning on the left-most cell, the probability of eventually falling off the left edge is* 1.

This is the famous *recurrence* phenomenon, because it asserts that such a random walk has probability one of eventually returning to its starting point. It follows that with probability one the random walk reaches any given fixed position of the tape. Interestingly, the recurrence property holds for random walks in dimensions one and two, but not in dimensions three or higher.

Putting everything together, let us show that B has asymptotic probability one. Fix any $\epsilon > 0$ and, by Lemma 2.5, find some large k such that with probability exceeding $\sqrt{1-\epsilon}$, the k-step random walk falls off the left edge of the tape. By Lemma 2.4, let n be large enough so that B_k contains more than the proportion $\sqrt{1-\epsilon}$ of all n-state programs. Combining these facts with the observation that programs in B_k operate statistically like random walks for their first k steps of computation (or until they halt, if this is sooner), as far as the head position is concerned, we conclude that proportion at least $(\sqrt{1-\epsilon})^2 = 1-\epsilon$ of all n-state programs exhibit the desired property. So the set B of all such programs has asymptotic probability one, and the theorem is proved. □

Let us now clarify matters by untangling the two possibilities for programs in B, namely, (1) the programs that halt before repeating a state and (2) the programs that fall off the tape before repeating a state. The fact is that behavior (1) is very rare and behavior (2) occurs with asymptotic probability one.

Theorem 2.7 *In the standard model, the asymptotic probability one behavior of a Turing machine, on any fixed input, is that the head falls off the tape before halting or repeating a state.*

Proof First, we generalize Lemma 2.4 to exclude the possibility of halting.

Lemma 2.8 *In the standard model, for any fixed input and fixed integer $k \geq 0$, the set of programs not repeating states and not halting within the first k steps of computation on that input has asymptotic probability one.*

Proof Let C_k be the desired set of programs, which includes the programs that fall off the tape within the first k steps of computation on that fixed input, provided that they do so before repeating a state. As in Lemma 2.4, we show inductively that C_k has asymptotic density one. When $k = 0$, this is trivial. Let us now calculate the probability that a program p is in C_{k+1}, given that it is in C_k. If the head fell off within k steps, then p will also be in C_{k+1}. Otherwise, as in Lemma 2.4, the first k steps of computation exhibit states $q_{i_0}, q_{i_1}, \ldots, q_{i_k}$, which have not yet repeated. The $(k+1)$th step of computation involves a transition $p(q_{i_k}, j_k) = (q_{i_{k+1}}, j_{k+1}, m_k)$, which will place p into C_{k+1} if $q_{i_{k+1}}$ is a new state and not the *halt* state. Since there are $n-(k+1)$ remaining states to choose from, the probability that p will be in C_{k+1} is at least $\frac{n-(k+1)}{n}$. Since this probability goes to 1 as n goes to infinity, we conclude that C_{k+1} has asymptotic probability one. □

Now fix any $\epsilon > 0$. Select k large enough so that the random walk in k steps has probability exceeding $\sqrt{1-\epsilon}$ of falling off the left edge. By Lemma 2.8, take n sufficiently large so that the proportion of all n-state programs that do not halt in k steps and do not repeat a state in k-steps is at least $\sqrt{1-\epsilon}$. Thus, as in the Main Theorem, these computations behave statistically like random walks, as far as the head position is concerned, and so the proportion of all n-state machines that fall off the tape in k steps before repeating a state or halting is at least $\sqrt{1-\epsilon}\sqrt{1-\epsilon} = 1-\epsilon$, as desired. □

Corollary 2.9 *In the standard model, the halting problem H has asymptotic probability zero, and the complement of H contains a decidable set of asymptotic probability one.*

Proof If the head falls off the tape, then the computation cannot reach the *halt* state, and so the program is not in H. So H has probability zero. The set of programs that fall off the tape before repeating a state or halting is contained in the complement of H, is clearly polynomial time decidable, and, by Theorem 2.7, has asymptotic probability one. □

The previous corollary depends on the formalism that computations for which the head falls off the tape are not counted as halting. If one wishes instead to count them as halting, then the conclusion would be that the corresponding version of H would have asymptotic probability one and contain a decidable set of asymptotic probability one.

Because the computational behavior identified in Theorem 2.7, with the head falling off the tape before a state is repeated, is both typical and trivial, many other well-known undecidability problems for Turing machines can also be decided with asymptotic probability one. We give two examples.

Definition 2.10 Let FIN be the set of programs computing functions on $\mathbb{N}$ with finite domain and COF be the set of programs with cofinite domain. These sets are well known to be undecidable (see, e.g., [3]).

Corollary 2.11 *In the standard model, there is a set B of programs such that*

(1) *B has asymptotic probability one,*
(2) *B is polynomial time decidable,*
(3) *$\mathrm{FIN} \cap B$ is polynomial time decidable,*
(4) *$\mathrm{COF} \cap B$ is polynomial time decidable.*

Proof For the purposes of computing functions on $\mathbb{N}$, we assume that input on a Turing machine tape is given by a string of 1s, that is, in unary form. Let B be the set of programs that fall off the tape before halting or repeating a state, on a tape initially filled entirely with 1s. This set is clearly polynomial time decidable, and by Theorem 2.7, it has asymptotic probability one. But any program that falls off the tape will have had a chance to inspect only finitely many of the 1s on the tape before doing so, and so the program will have this same behavior provided that there is a sufficiently long string of 1s on the tape as input. So every program in B is in FIN and none are in COF. On B, therefore, these questions are decidable. □

The previous proof shows that almost every program computes a finite function. In other words, FIN has asymptotic probability one and COF has asymptotic probability zero. Taking the domains of the computable functions as the natural enumeration of the computably enumerable (c.e.) sets, this means that almost every c.e. set is finite.

The general conclusion that we make from the main theorem and these corollaries is that the central concerns of computability theory and complexity theory do not interact well with the natural measure, and the most important phenomena occur on sets of measure zero. This is a negative conclusion, since it shows that one should not expect interesting complexity issues to arise with probability one.

We resist the argument that the Turing machine programs in our set B of the main theorem are "meaningless" programs that should not be counted as within the domain of discourse. First, we note that programs in B may have better behavior on other inputs, and so it would be wrong to exclude the programs entirely, based merely on their behavior on the input 0; we do not yet know whether programs have probability one of falling off the tape on *all* their input. Second, we observe that the programs are not meaningless, for they each determine a well-defined computational behavior; it is just that with probability one, this behavior is trivial in a certain way. Third, although the programs do fall off the tape, at least some of them leave behind computationally interesting information on the tape, and so it would be incorrect to characterize them as being uniformly noncomputational. Finally, we believe that the space of all Turing machine programs is defined with very natural boundaries—any syntactically correct complete sequence of instructions is allowed—and we would find it unnatural to define the program space with any except a purely syntactic criteria.

3 Other Computational Models

Let us turn now to the question of whether the conclusions of the main theorem hold for other models of Turing machine computability. First, we observe that the argument of the main theorem applies directly to several other common Turing machine models.

Corollary 3.1 *The conclusion of the main theorem holds for the following models of computability.*

(1) *Single tape Turing machines with an arbitrary finite alphabet, operating on a semi-infinite tape.*

(2) *Multitape Turing machines with an arbitrary finite alphabet, operating on semi-infinite tapes.*

(3) *Turing machines with an arbitrary finite alphabet, operating with a head moving on a half-plane or quarter-plane grid of cells.*

Proof The corollary is proved merely by observing that the calculations of Lemma 2.4 do not fundamentally rely on the size of the alphabet, and so in the case of a general alphabet, it is still true that for any k the set of programs that adopt new states for their first k moves has asymptotic probability one. Because these programs therefore act like a random walk for the first k steps, the probability that they fall off the left end of the tape can be made as close to 1 as possible. So (1) holds. For the multitape model of (2), we assume that there is a single head moving back and forth, reading and writing on all columns at once. This is functionally equivalent to having a larger alphabet, if one regards an entire column of cell values as a single element of a larger alphabet. So (2) holds. For (3), we observe that first, the analogue of Lemma 2.4 remains true, and second, the desired conclusion now follows from the two-dimensional generalization of Lemma 2.5, by which random walks on a half-plane (or any smaller portion of the plane), eventually fall off the edge with probability one. □

The result also applies to 2-dimensional Turing machines operating on a full doubly-infinite plane, provided that it has at least one broken cell, which is broken in the sense that it causes the computation to cease if the head should happen to occupy it. The point is that because of the 2-dimensional analogue of Polya's recurrence theorem, on any fixed input such a Turing machine would, with asymptotic probability one, land on the forbidden cell before repeating a state.

The reader will have already observed that our argument does *not* work, of course, with the computational models having bi-infinite tapes, where there is no possibility of the head falling off. Nevertheless, by means of a totally different argument, we will now establish the conclusions of the main theorem for at least some of the bi-infinite tape models.

Specifically, consider the *halting subset* model of computation, which augments any of the usual Turing machine models by specifying not a single distinguished *halt* state, but rather by specifying a subset of the states to be halting states. In this model, we have a finite set of states $Q = \{q_1, \ldots, q_n\}$, and a program is specified by providing a transition function $p : Q \times \{0, 1\} \to Q \times \{0, 1\} \times \{L, R\}$ and also by listing a subset $A \subseteq Q$ of the states, decreed to be the *halting* states. Computation proceeds in the ordinary manner until a state is reached that is an element of the set A of halting states, at which point the computation halts. This model of computation works equally well with bi-infinite or semi-infinite tapes (or planar or multidimensional tapes, as desired, with suitable modifications to the instruction set to allow for appropriate head movement).

Theorem 3.2 *The conclusion of the main theorem holds for the halting subset model of Turing machine computation using any sort of tape configuration and any finite alphabet.*

Proof The idea of this proof is that since each state has a 50% chance of being a halting state, it is very likely for the computation to halt very quickly. For definiteness, let us consider initially the bi-infinite tape halting subset model. Let B be the set of programs that halt before repeating a state on input 0. This is clearly a polynomial time decidable set, and the halting problem for programs in B is easily decided (because all programs in B halt). It remains to show that B has asymptotic probability one. For any natural number k, the arguments of the main theorem, and specifically Lemma 2.4, show that there is a measure one set of programs that do not repeat states for the first k steps of computation (or halt before k steps without having repeated a state). Among the programs that do not repeat states for k steps, each new state has an independent 50% chance of being a halting state. So the chance of halting in these k steps, given that no states have been repeated, is $1-2^{-k}$. Thus, the asymptotic probability of B exceeds $1-2^{-k}$ for every k, and so it has measure one, as desired. The argument works just as well with the halting subset models of computation using other tape configurations by modifying B to include all programs that either halt or cease computation before repeating a state. The essential calculation above shows that this set has asymptotic measure one, is polynomial time decidable, and for programs in this set, the halting problem is polynomial time decidable, as desired. □

Unfortunately, we do not know whether the conclusions of the main theorem hold for the relatively common model of Turing machine computation having a bi-infinite tape and a single halt state. Neither do we know whether it holds for the semi-infinite tape models that allow computation somehow to continue after attempting to move left from the left-most cell. We admit that this situation is unsatisfactory, because one doesn't like results in computability theory to be sensitive to the choice of computational model.

Question 3.3 *Does the conclusion of the Main Theorem hold for all the usual models of Turing machine computation?*

The current focus, of course, is on the bi-infinite tape model with a single halt state. One can weaken the desired conclusion by asking only that the halting problem be decided on a set of large probability, rather than probability one. If one asks only to decide the problem on a set of nonzero probability, then this is a consequence of Lemma 2.2.

Theorem 3.4 *For any of the single halt state models of Turing machine, including those with bi-infinite tapes, there is a set B of Turing machine programs such that*

(1) *B has nonzero asymptotic probability,*
(2) *B is polynomial time decidable,*
(3) *the halting problem $H \cap B$ is polynomial time decidable.*

Proof The set of programs arising in Lemma 2.2, which have no transition leading to the *halt* state, has asymptotic probability $1/e^2$, but clearly no such program is in H. □

Question 3.5 *In Theorem 3.4, how large can the probability of B be? Can one always decide the halting problem in an asymptotic majority of cases?*

We close the paper by emphasizing that the relation of Turing equivalence does not respect asymptotic probability.

Theorem 3.6 *The relation of Turing equivalence does not respect the property of having asymptotic probability one in the natural numbers. Indeed, for any set A of natural numbers there is a set $B =_T A$ that is Turing equivalent to A and has any prescribed asymptotic computable probability or nonconvergent probability.*

Proof If a set A has asymptotic probability one in the natural numbers, then the complement of A, which is Turing equivalent to A, has asymptotic probability zero. But also, any set A is Turing equivalent to a set with asymptotic probability zero by simply multiplying its second member by 2, its third member by 3, and so on, so as to stretch it out to density zero. The complement of this set, which is also Turing equivalent, has asymptotic probability one. Intermediate densities can be achieved by adding regular blocks of numbers in a computable pattern so as to achieve a given intermediate computable probability while the true information is coded on a thin set of probability zero. By alternating blocks of numbers with large empty stretches, one can arrange that the asymptotic probability of the set does not converge and even that the upper density is 1 while the lower density is 0 (or any other intermediate values). Meanwhile, the true information of the set is coded on a thin set, of probability zero, which does not upset those calculations. □

Finally, we note that with many models of computability, the notion of what happens "almost everywhere" can be highly sensitive to what are otherwise unimportant differences in formalism. For example, if one takes as the basic model of computability a suitable generalization of C++ programs, then most would agree that for the usual purposes it is an irrelevant formalism whether one excludes programs at the outset that have syntax errors preventing them from compiling or instead takes them to compute the empty function. But if they were officially counted, then because clearly there are far more programs with errors than without, it would mean that almost every program would be trivial in this way. For such a model, all interesting phenomena would occur on a set of asymptotic density zero.

References

[1] Feller, W., *An Introduction to Probability Theory and Its Applications. Vol. I*, 3d edition, John Wiley & Sons Inc., New York, 1968. Zbl 0155.23101. MR 0228020. 518

[2] Pólya, G., "Über eine Aufgabe der Wahrscheinlichkeitsrechnung betreffend die Irrfahrt im Straßennetz," *Mathematische Annalen*, vol. 84 (1921), pp. 149–60. Zbl 48.0603.01. MR 1512028. 518

[3] Soare, R. I., *Recursively Enumerable Sets and Degrees. A Study of Computable Functions and Computably Generated Sets*, Perspectives in Mathematical Logic. Springer-Verlag, Berlin, 1987. Zbl 0667.03030. MR 882921. 519

Acknowledgments

The research of the first author has been supported in part by grants from the CUNY Research Foundation.

Department of Mathematics
The Graduate Center of The City University of New York
365 Fifth Avenue
New York NY 10016

Department of Mathematics
The College of Staten Island of CUNY
Staten Island NY 10314
jhamkins@gc.cuny.edu
http://jdh.hamkins.org

McGill University
Department of Mathematics and Statistics
Burnside Hall
805 Sherbrooke West
Montreal QC
CANADA H3A 2K6
alexeim@math.mcgill.ca
http://www.math.mcgill.ca/~alexeim/home.php