Noisy Turing Machines

Eugene Asarin[1] and Pieter Collins[2]

[1] LIAFA, Université Paris 7 / CNRS, case 7014, 2 place Jussieu, 75251 Paris Cedex 05, France, Eugene.Asarin@liafa.jussieu.fr
[2] Centrum voor Wiskunde in Informatica, P.O. Box 94079, 1090 GB Amsterdam, The Netherlands, Pieter.Collins@cwi.nl

Abstract. Turing machines exposed to a small stochastic noise are considered. An exact characterisation of their ($\approx \Pi_2^0$) computational power (as noise level tends to 0) is obtained. From a probabilistic standpoint this is a theory of large deviations for Turing machines.

1 Introduction

Computers are always subjected to faults and component failures, and even random changes of memory bits caused by to cosmic rays or neutrons flipping memory cells [1]. From the practical viewpoint these phenomena are particularly important for computers operating in hostile environments, such as aboard a spacecraft [2]. In the present paper we adopt a more theoretical and abstract approach to this issue and study how small random perturbations can affect the computational power of popular computational models, in our case Turing machines (TMs).

As far as we know, the pioneering paper considering influence of infinitesimal noise on computational models was Puri's [3], where the author introduces the infinitesimally perturbed semantics and solves the reachability problem for timed automata. Fränzle [4] applies a version of Puri's noise to hybrid systems, and argues that such a noise leads to a sort of "practical decidability". The immediate predecessor of the present paper is [5] where computational power is analysed for infinitesimally perturbed hybrid systems and TMs. The main result of [5] is the Π_1^0 completeness of reachability or acceptance problems for such machines. It is important to notice that all the papers cited above considered a non-deterministic noise of a bounded (and tending to zero) magnitude, with no probabilistic assumptions.

The influence of a small stochastic noise on computational models has been considered in [6] for finite-state models, and in [7] for neural networks. Other related work concerns the behaviour of dynamical systems under a small stochastic noise, known as the theory of large deviations. A good reference is [8].

In the present paper we consider TMs exposed to a small stochastic noise, or in other words large deviations for TMs. We give an exact characterisation of their computational power in terms of classes of arithmetic hierarchy (see [9]).

The rest of the paper is organised as follows. In Sec. 2 we introduce Noisy Turing Machines (NTMs) and several versions of "computability" by such machines. In Sec. 3 we explore NTMs with a noise level $\varepsilon > 0$ and establish some basic properties, such as continuity and computability of acceptance probability, decidability of the halting problem etc. In Sec. 4 we describe several interesting NTMs used in the subsequent sections. This section also gives a flavour of "noisy programming" and reasoning about noisy programs. The main technical results of the paper are established in Sec. 5 and 6 where we explore the computational power of NTMs for a noise level *tending to zero*. Such a "limit computational power" turns out to be stronger than that of a TM; we give its precise characterisation.

2 The Model

We consider a standard multi-tape TM augmented by an additional parameter giving the noise level. Formally, a *Noisy Turing Machine* (NTM) is a tuple

$$\mathcal{M}_\varepsilon = (Q, \Sigma, \Gamma, N, \rho, \varepsilon, q_0, q_\top, q_\bot)\,, \tag{1}$$

where Q is the set of states, Σ is the input alphabet (not containing the special blank symbol $\sqcup$), $\Gamma \supset \Sigma \cup \{\sqcup\}$ is the tape alphabet, N is the number of tapes, $\rho : Q \times \Gamma^N \to Q \times \Gamma^N \times \{L, R, S\}^N$ is the transition function, ε is the noise level, q_0 is the initial state, $q_\top$ the accepting state and $q_\bot$ the rejecting state.

Every tape is infinite in both directions. Initially the ith tape contains a word $w_i \in \Sigma^*$ completed by two infinite tails of blank symbols $\sqcup$, and the ith tape head is pointing to the first symbol of w_i. Every computation step performed by an NTM consists of two stages:

At the *noisy stage* the tapes are exposed to a noisy environment, which changes each symbol on the tape independently with probability $\varepsilon \ll 1$. A changed symbol takes any other value in the tape alphabet with equal probability.

At the *progress stage* the computation proceeds as follows. Starting in state q^-, the machine reads the symbol on each tape, giving an N-tuple $s^- \in \Gamma^N$. If $\rho(q^-, s^-) = (q^+, s^+, m^+)$, the machine changes to state q^+, writes $(s^+)_i$ on the ith tape, and shifts the ith tape head left if $(m^+)_i = L$, right if $(m^+)_i = R$, and does not move it if $(m^+)_i = S$. Whenever the machine arrives at $q_\top$ or at $q_\bot$ it halts.

We are interested in the probabilities $\mathbb{P}\left(\mathcal{M}_\varepsilon(w)\downarrow\right)$, $\mathbb{P}\left(\mathcal{M}_\varepsilon(w) = \top\right)$ and $\mathbb{P}\left(\mathcal{M}_\varepsilon(w) = \bot\right)$ that, for a given noise level ε and a given input word w, the NTM $\mathcal{M}$ halts, accepts or rejects, respectively. We are even more interested in the behaviour of those probabilities as $\varepsilon \to 0$.

Definition 1. *An NTM $\mathcal{M}$ is*

- lim-halting *if $\forall x \in \Sigma^*$, $\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x)\downarrow\right) = 1$, i.e the limit probability to halt is 1;*
- almost sure (a.s.)-halting *if for any $x \in \Sigma^*$ and any $\varepsilon > 0$ the probability to halt is* 1*;*
- converging *if $\forall x \in \Sigma^*$, $\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right)$ exists.*

Clearly if an NTM is a.s.-halting, then it is lim-halting. The former two properties seem restrictive, but in Sec. 3 it will be shown that any NTM is equivalent to an a.s.-halting one, in the sense that the acceptance probabilities are equal.

The "limit computational power" of an NTM captures the behaviour of a machine operating in an environment which is almost, but not entirely, noise-free.

Definition 2. *An NTM $\mathcal{M}$* lim-*generates a function $p : \Sigma^* \to [0,1]$ if it is* lim-*halting, converging, and for any $x \in \Sigma^*$, the limit probability to accept it is $p(x)$:*

$$\forall x \in \Sigma^*,\ \lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right) = p(x).$$

An NTM $\mathcal{M}$ lim-*decides a set $S \subset \Sigma^*$ if it* lim-*generates its characteristic function.*

Notice that in order to lim-decide a set S, an NTM should satisfy a 0-1 law:

$$\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right) = \begin{cases} 1, \text{ if } x \in S; \\ 0, \text{ if } x \notin S. \end{cases}$$

A weaker notion of computability considering lim sup (or lim inf) rather than lim is suitable for non-converging machines.

Definition 3. *An NTM $\mathcal{M}$* lim sup-*generates a function $p : \Sigma^* \to [0,1]$ if it is* lim-*halting and for any $x \in \Sigma^*$, the upper limit probability to accept it is $p(x)$:*

$$\forall x \in \Sigma^*,\ \limsup_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right) = p(x).$$

An NTM $\mathcal{M}$ lim sup-*decides a set $S \subset \Sigma^*$ if for any $x \in \Sigma^*$ it* lim sup-*generates its characteristic function.*

The question whether this can be really considered as a computation is left to a philosophically-minded reader.

2.1 If it Halts without Noise

We start the study of noisy machines with the easy case when a machine without noise halts on an input x.

Theorem 1. *If $\mathcal{M}(x) = \top$ then $\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right) = 1$. Symmetrically, if $\mathcal{M}(x) = \bot$ then $\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \bot\right) = 1$.*

Proof. Consider the first case, $\mathcal{M}(x) = \top$; the other case is similar. Let τ be the computation time of $\mathcal{M}$ on the input x. For any $\delta > 0$ take an $\varepsilon < \delta\tau^{-2}$.

A normal computation (without noise) of $\mathcal{M}$ on x uses at most τ tape cells during τ time units. When noise is added the probability for at least one of those cells being perturbed during τ time units cannot exceed $\tau \cdot \tau \cdot \varepsilon < \delta$. The computation of $\mathcal{M}$ on x leading to acceptance is then unaffected by the noise, hence the NTM $\mathcal{M}_\varepsilon$ accepts x with probability at least $1-\delta$. Since δ is arbitrary, $\lim_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right) = 1$. □

Corollary 1. *If a TM $\mathcal{M}$ decides a set $S \subset \Sigma^*$ then its NTM version* lim-*decides the same set S.*

3 General Properties: $\varepsilon > 0$

In this section we explore NTMs and acceptance by NTMs for a positive noise level $\varepsilon > 0$. This paves the way for the characterisation of the limit behaviour of NTMs as $\varepsilon \to 0$ in subsequent sections.

3.1 Automaton

For an NTM $\mathcal{M}$ described by a tuple (1), we can abstract away the memory tapes to obtain an automaton

$$\mathcal{A} = (Q, \rho', q_0, q_\top, q_\bot)\,,$$

where the transition relation $\rho' \subset Q \times Q$ is obtained from $\rho : Q \times \Gamma^N \to Q \times \Gamma^N \times \{L, R, S\}^N$ by projection. Any allowable sequence of transitions of $\mathcal{A}$ is a possible sequence of transitions for $\mathcal{M}$, since it is always possible that the element at the tape head changes just before the step to enable the desired transition.

We say a state q of $\mathcal{M}$ is a *looping state* if for every possible sequence of transitions starting at q, no halting state is reached. In this case, the probability that $\mathcal{M}$ halts given that it reaches q is zero.

Theorem 2. *For any NTM $\mathcal{M}$, there exists an effectively constructible a.s.-halting NTM $\mathcal{M}'$ with the same acceptance probability for any input $x \in \Sigma^*$ and any $\varepsilon > 0$.*

Proof. We construct $\mathcal{M}'$ by deleting all looping states from $\mathcal{M}$, and replacing all transitions leading to looping states with a transition leading to $q_\bot$. Then clearly, $\mathbb{P}\left(\mathcal{M}'_\varepsilon(x) = \top\right) = \mathbb{P}\left(\mathcal{M}_\varepsilon(x) = \top\right)$.

It remains to show that $\mathbb{P}\left(\mathcal{M}'_\varepsilon(x)\downarrow\right) = 1$. Since $\mathcal{M}'$ has no looping states, for any state q of $\mathcal{M}'$, there is a sequence of transitions leading to a halting state in at most $k = |Q|$ steps. This sequence of transitions occurs for the noisy machine with probability at least $(\varepsilon/|\Sigma|)^k$. Therefore the probability that the machine halts after nk steps is at least $1 - \left(1 - (\varepsilon/|\Sigma|^k\right)^n$. Hence $\mathcal{M}'$ halts with probability 1. □

We can therefore replace any NTM with one that halts with probability 1 for every $\varepsilon > 0$ without changing the acceptance probability. This means that, unlike ordinary TMs, we need only consider NTMs which almost surely halt on any input. In the rest of this paper, we assume that all NTMs have no looping states.

3.2 Continuity

Theorem 3. $\mathbb{P}\left(\mathcal{M}_\varepsilon(w) = \top\right)$ *is continuous with respect to ε for $\varepsilon > 0$.*

Proof. Let $p(w, \varepsilon, t)$ be the probability that $\mathcal{M}$ halts in time t in state $q_\top$, and $q(w, \varepsilon, t)$ be the probability that $\mathcal{M}$ halts in time t in state $q_\perp$. Then clearly $p(w, \varepsilon, t)$ and $q(w, \varepsilon, t)$ are continuous as functions of $\varepsilon > 0$, since they depend on finite computations. Let $r(w, \varepsilon, t) = 1 - p(w, \varepsilon, t) - q(w, \varepsilon, t)$. Then by our standing assumption of almost-sure halting, $r(w, \varepsilon, t) \to 0$ as $t \to \infty$.

It is easy to see that

$$p(w, \varepsilon, t) < \mathbb{P}(\mathcal{M}_\varepsilon(w) = \top) < p(w, \varepsilon, t) + r(w, \varepsilon, t) = 1 - q(w, \varepsilon, t). \quad (2)$$

To prove continuity of $\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top)$ at ε, take t such that $r(w, \varepsilon, t) < \delta/3$. For ε' sufficiently close to ε, both $|p(w, \varepsilon', t) - p(w, \varepsilon, t)| < \delta/3$ and $r(w, \varepsilon', t) < 2\delta/3$. Then, using (2), we obtain $|\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top) - \mathbb{P}(\mathcal{M}_{\varepsilon'}(w) = \top)| < \delta$. Hence $\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top)$ is continuous. □

3.3 Computability

We now consider ε-perturbed machines for a fixed rational ε. By computability of a real number x (see [10]), we mean that given an error bound $\delta > 0$, there is a TM which computes an approximation to x with an error of at most δ.

Theorem 4. $\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top)$ *is computable as a function of* $\mathcal{M}$*, rational* $\varepsilon > 0$ *and* w*.*

Proof. Let $p(w, \varepsilon, t)$, $q(w, \varepsilon, t)$ and $r(w, \varepsilon, t)$ be as in the proof of Theorem 3. By simulating all possible runs of the NTM of length at most t and computing the probability of each, we can compute p, q and r exactly. Since $r(w, \varepsilon, t) \to 0$ as $t \to \infty$, we can take t sufficiently large so that $r(t, \varepsilon, t) < \delta$, and so $|\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top) - p(w, \varepsilon, t)| < \delta$. □

4 Some Gadgets

We now describe some generally useful NTMs and their properties.

4.1 Measuring Time

The first gadget is a Timer. Its construction is very simple: it is just a TM with one tape (initially blank), whose head goes right at every step. If it sees a non-blank cell it stops.

The following lemma establishes that, when subjected to an ε-noise, Timer is capable to measure approximately a lapse of $\varepsilon^{-1/2}$ time units.

Lemma 1. *Let* τ *be the time before the* Timer$_\varepsilon$ *stops. Then for any* a, b *with* $a < 1/2 < b$*, the following estimates hold:*

1. $\mathbb{P}(\tau < \varepsilon^{-a}) = O(\varepsilon^{1-2a})$;
2. $\mathbb{P}(\tau > \varepsilon^{-b}) = O(\varepsilon^{d})$ *for any* $0 < d < 1/2$.

Proof. First we estimate the probability of the event E_1 that $\tau < \varepsilon^{-a}$. This probability can be majorated by the probability of the event E_2 that during $\lfloor \varepsilon^{-a} \rfloor$ time units at least one of the first $\lfloor \varepsilon^{-a} \rfloor$ cells on the tape has been altered by the noise. For each cell and each step the probability to be altered is ε, which gives an upper bound

$$\mathbb{P}(E_1) \leq \mathbb{P}(E_2) \leq \lfloor \varepsilon^{-a} \rfloor \cdot \lfloor \varepsilon^{-a} \rfloor \cdot \varepsilon = O(\varepsilon^{1-2a}).$$

In the sequel we will omit $\lfloor \cdot \rfloor$ and $\lceil \cdot \rceil$ symbols and make all the computations as if all the powers of ε considered were integer numbers.

The event E_3 that $\tau > \varepsilon^{-b}$, implies either the event E_4 that none of the first ε^{-b} cells have been modified before the timer scans them, or the event E_5 that at least one of the first ε^{-b} cells has been modified at least twice in time ε^{-b}. Hence $\mathbb{P}(E_3) \leq \mathbb{P}(E_4) + \mathbb{P}(E_5)$.

E_4 is a conjunction of $\varepsilon^{-b} \cdot \varepsilon^{-b}/2$ independent events with probabilities $1-\varepsilon$; each event means that a cell has not been perturbed at a time instant. Hence

$$\mathbb{P}(E_4) = (1-\varepsilon)^{\varepsilon^{-2b}/2} = O(\exp(-\varepsilon^{1-2b})).$$

In particular, if $b > 1/2$, $\mathbb{P}(E_4) = o(\varepsilon^n)$ for any $n > 0$.

The event E_5 is a disjunction of independent $n = \varepsilon^{-b}$ events. Each of those events is that a particular cell has been perturbed at least twice during ε^{-b} time units. Hence

$$\mathbb{P}(E_5) \leq n\left(1 - (1-\varepsilon)^n - n\varepsilon(1-\varepsilon)^{n-1}\right) = O(n^3\varepsilon^2) = O(\varepsilon^{2-3b}).$$

Therefore, $\mathbb{P}(E_3) = O(\varepsilon^{2-3b})$ for $1/2 < b < 2/3$. Since $\mathbb{P}(E_3)$ is a decreasing function of b, we must have $\mathbb{P}(E_3) = O(\varepsilon^d)$ for any $0 < d < 1/2$. □

We say an event E occurs with *high probability* if there exists $d > 0$ such that $\mathbb{P}(E) = 1 - O(\varepsilon^d)$. Similarly, it occurs with *low probability* if $\mathbb{P}(E) = O(\varepsilon^d)$.

We remark that it would be easier to build a timer measuring ε^{-1} lapse of time. Such a timer is an NTM staying in place and observing one cell, until its contents is modified. Unfortunately, such a timer would be rather useless, because during such a long time the contents of all the cells on the tape becomes completely random.

We prefer the $\varepsilon^{-1/2}$ timer described at the beginning of this section because during such time the probability of perturbation on a small zone of tape or of a "double error" in the same position of two tapes is low. In the next subsection we formalise these properties, and explain how they allow for reliable computations of duration $\varepsilon^{-1/2}$ and even more. We will then be able to use these constructions to build $O(\varepsilon^{-c})$ TIMERS for $1/2 < c < 1$.

4.2 Tossing Coins

By letting two TIMERS race each other to find a non-blank symbol, we can generate random bits.

A RandomBit machine has two tapes. Two Timers are launched concurrently on both tapes. If the first one stops before the second one, the result is $\perp$, if the second one stops before the result is $\top$. In the highly improbable case of a tie, the Timers are restarted. The following result is straightforward, but important; it shows that NTMs can produce random bits.

Lemma 2. *The* RandomBit$_\varepsilon$ *terminates almost surely and returns $\perp$ and $\top$ (or 0 and 1) with probabilities $1/2$ each. Its computation time is bounded above by ε^{-b} (with $b > 1/2$) with probability $1 - O(\varepsilon^d)$ for any $0 < d < 1/2$.*

Notice that the RandomBit$_\varepsilon$ gadget can also be started some time T after the beginning, and can be run continuously to generate a succession of random bits.

4.3 Memory

Even on time scales of order $\varepsilon^{-1/2}$ generated by a Timer, the behaviour of a noisy version of a regular TM $\mathcal{M}$ has unacceptably high errors. To obtain correct execution with high probability of a time interval of order ε^{-a}, we run all computations of $\mathcal{M}$ on a multi-tape Memory with error correction.

For computations taking $O(\varepsilon^{-a})$ time with $a < 2/3$ we can guarantee error-freedom with high probability by taking three identical copies of the tape and making the same computation on all of them. If at some moment a disagreement between the three tapes is observed, it is immediately corrected by a majority vote. This procedure allows to correct single errors, while Lemma 3 ensures that double errors are highly improbable.

By using more tapes, we can, in fact, construct Memory which is error-free with high probability on time intervals $O(\varepsilon^{-c})$ for any $c < 1$.

Lemma 3. *Let $\mathcal{M}_\varepsilon$ be an NTM running on a three-tape* Memory *for a time period $O(\varepsilon^{-a})$ using space $O(\varepsilon^{-b})$. Then the probability of incorrect execution of $\mathcal{M}_\varepsilon$ is of order $O(\varepsilon^{2-2a-b})$.*

Proof. Incorrect execution can only occur if two cells with the same coordinate are perturbed in time period $\tau = O(\varepsilon^{-a})$. The probability of such a "double error" in a given cell is $O(\varepsilon^{2(1-a)})$, hence the probability of a double error in any of b cells is $O(\varepsilon^{2-2a-b})$.

4.4 Programming NTMs

Armed with a Timer, a RandomBit and (fairly) reliable Memory, we can start to program NTMs by running ordinary TMs on a Memory, using a Timer to halt the computation before the noise-induced errors become too high.

A simple, but very useful gadget is a Counter machine. This machine stores an integer n in binary form in a three-tape Memory. The Counter spends all its time incrementing its value, which asymptotically grows as $n \sim t/\log t$.

Using a Counter, we can construct a Delay gadget. When this gadget is activated, it copies the time n contained in the counter, and computes some (easy to compute) function $f(n) \sim n^s$. It then waits until the COUNTER reaches $f(n)$, emits a signal, and goes back to sleep.

Using the COUNTER and DELAY gadgets, we can construct an improved version of a TIMER. We run an $\sim \varepsilon^{-1/2}$ timer as usual, but when this stops, we activate a DELAY gadget with $f(n) \sim n^{2c}$. The program continues running until the DELAY gadget deactivates. This new TIMER(c) gadget stops in time $\sim \varepsilon^{-c}$ with high probability.

We can use TIMER(c) and RANDOMBIT to construct a RANDOMNUMBER. By storing successively generated bits in a MEMORY, we generate an increasing sequence of rationals r_i converging to a real number r which is uniformly distributed in $[0, 1]$. By using a TIMER(c), we can generate $\sim \varepsilon^{1/2-c}$ digits of r.

4.5 Oscillators

An OSCILLATOR is a gadget which stores a binary digit in a "register" variable where it is unaffected by the noise. (Formally, we construct a register by taking a set of states $Q \times \{0, 1\}$.) When the OSCILLATOR is halted by a TIMER, it stops in $q_\top$ if the register holds 1, and in $q_\bot$ if the register holds 0.

A simple oscillator which changes register state at every step is not very interesting; the limiting acceptance probability is 1/2. By using a DELAY, we can hold the value of the register for a period $[m, f(m) \sim m^c]$; long enough for its value to be seen when the TIMER halts.

Lemma 4. *Let $\mathcal{M}$ be an* OSCILLATOR *which uses a delay to switch state at times $n_i = f(n_{i-1})$ with $f(n) \geq n^c$, and which halts when a* TIMER *stops. Then M halts almost surely, but $\mathbb{P}(\mathcal{M}_\varepsilon = \top)$ does not converge as $\varepsilon \to 0$.*

Proof. Choose a, b such that $a < 1/2 < b < 2/3$, $b/a < c$ and $a + b < 1$, and let $d = 1 - a - b$. For any given n, we can find $\varepsilon < 1$ such that $[\varepsilon^{-a}, \varepsilon^{-b}] \subset [n, n^c]$. If oscillator switches at times n_i, then $n_{i+1} \geq n_i^c$. Hence, there is a sequence ε_i with $\varepsilon_i \to 0$ as $i \to \infty$ such that $[\varepsilon_i^{-a}, \varepsilon_i^{-b}] \subset [n_i, n_{i+1}]$.

The TIMER halts at $\tau \in [\varepsilon_n^{-a}, \varepsilon_n^{-b}]$ with high probability. Further, in this time the MEMORY ensures correct execution with high probability. Hence for noise levels ε_{2i}, the probability that the timer halts with the register in state 0 tends to 1 as $n \to \infty$, and for noise levels ε_{2i+1} timer halts with the register in state 1 with high probability. Thus $\mathbb{P}(\mathcal{M}_\varepsilon = \top)$ oscillates between 0 and 1 as $\varepsilon \to 0$, and does not converge as $\varepsilon \to 0$. □

5 Decisional Power of NTMs

In this section we address the capabilities of NTMs for deciding sets $S \subset \Sigma^*$. The main result of this section is

Main Result 1. *A set $S \subset \Sigma^*$ is* lim-*decidable if and only if it is Δ_2^0. A set $S \subset \Sigma^*$ is* lim sup-*decidable if and only if it is Π_2^0.*

The upper complexity bounds follow from Theorems 9 and 11 in the next section.

5.1 Deciding Recursively Enumerable Sets

The following result illustrates how converging NTMs can be stronger than ordinary TMs, and solve, for example, the halting problem.

Theorem 5. *For any recursively enumerable (Σ_1^0) set S there exists an NTM $\mathcal{N}$ which* lim-*decides S.*

Proof. Recall that S is Σ_1^0 if there is a TM $\mathcal{M}$ such that $\mathcal{M}(w)$ halts if, and only if, $w \in S$. Given such a TM, we construct an NTM $\mathcal{N}_\varepsilon$ with 4 tapes to lim-decide S. On tapes 1, 2 and 3 we have a Memory store on which we run $\mathcal{M}$, correcting errors by majority vote. On tape 4, we run a Timer gadget.

The computation terminates in the accepting state if $\mathcal{M}$ runs successfully and reaches its halting state. The computation terminates in the rejecting state if the Timer stops.

Since the Timer stops almost surely, and does so with high probability in time $\tau \in [\varepsilon^{-a}, \varepsilon^{-b}]$, the computation performed by $\mathcal{M}$ runs successfully with high probability, terminating in state $q_\top$ if $w \in S$ (as long as ε is small enough, namely such that ε^{-a} exceeds the computation time of $\mathcal{M}(w)$), and halts in state $q_\perp$ if $w \notin S$. □

5.2 Deciding Δ_2^0 Sets

We now strengthen the result of Theorem 5 to show that NTMs can lim-decide Δ_2^0 sets. Recall that a set S is Δ_2^0 if both S and its complement are Π_2^0.

A particularly useful characterisation of a Π_2^0 set, similar to Büchi acceptance by ω-automata, can be given in terms of *signalling* TMs. A signalling TM has no halting states, but instead a distinguished set of *signalling states* $Q_s \subset Q$. A set S is Π_2^0 if there is a signalling TM $\mathcal{M}$ which enters states from Q_s infinitely often if, and only if, $w \in S$.

Theorem 6. *For any Δ_2^0 set S there exists an NTM which* lim-*decides S.*

Proof. Let $\mathcal{M}^1$ be a TM such that $\mathcal{M}^1(w)$ emits a signal infinitely often iff $w \in S$, and $\mathcal{M}^0$ a TM such that $\mathcal{M}^0(w)$ emits a signal infinitely often iff $w \notin S$.

We can run $\mathcal{M}^1$ and $\mathcal{M}^0$ in parallel with a register variable. Whenever $\mathcal{M}^i$ emits a signal, we store i in the register. If $w \in S$, then eventually $\mathcal{M}^0$ emits no more signals, but $\mathcal{M}^1$ continues to do so, and the register sticks to 1. Conversely, if $w \notin S$, then eventually the register contains 0. In both cases, after some time $T(w)$ the register's content never changes and is equal to $\chi_S(w)$.

We lim-decide S by an NTM $\mathcal{N}$ running $\mathcal{M}^1$ and $\mathcal{M}^0$ as described above. Computation is terminated when a Timer stops, the outcome is the register's content. Whenever ε is small enough to ensure that the Timer stops after time $T(w)$, we can easily see that $\mathcal{N}_\varepsilon$ operates correctly with high probability and outputs $\chi_S(w)$. Hence $\mathcal{N}$ lim-decides S. □

5.3 Deciding Π_2^0 Sets

We now investigate the computational power of NTMs without restriction of convergence.

Theorem 7. *For any Π_2^0 set S there exists an NTM which* lim sup-*decides S.*

Proof. Let $\mathcal{M}$ be a TM such that $\mathcal{M}(w)$ emits a signal infinitely often iff $w \in S$. To ensure that signals are noticed in the presence of noise, we combine $\mathcal{M}$ with a DELAY which activates when $\mathcal{M}$ emits a signal, setting a register to 1. Whenever DELAY deactivates, it resets the register to 0. As usual, the computation is halted whenever the TIMER stops.

Assuming correct operation of $\mathcal{M}$ and DELAY, which occurs with high probability, the register variable is equal to 1 on time intervals $[\tau_i, \tau_i^c]$ infinitely often if $w \in S$, and is eventually equal to 0 if $w \notin S$. Using the argument from the proof of Lemma 4 we can see that $\mathcal{N}$ lim sup-decides S. □

6 Generating Probability Functions

In this section, we investigate the functions which can be generated as the acceptance probability of an NTM as $\varepsilon \to 0$. We shall prove the following result:

Main Result 2. *A function $p : \Sigma^* \to [0;1]$ can be* lim-*generated by a converging NTM if and only if it is $0'$-computable. A function $p : \Sigma^* \to [0;1]$ is* lim sup-*generated by an NTM if and only if it is upper $0'$-semicomputable.*

6.1 Generating $0'$-Computable Probabilities

Recall that a function is called $0'$-computable if it can be computed by a TM using an oracle for the halting problem. Equivalently, p is $0'$-computable iff $\{(r, w) \in \mathbb{Q} \times \Sigma^* \mid r < p(w)\}$ and $\{(r, w) \in \mathbb{Q} \times \Sigma^* \mid r > p(w)\}$ are Π_2^0-sets.

Theorem 8. *Let $p : \Sigma^* \to [0, 1]$ be a $0'$-computable function. There exists a converging NTM $\mathcal{S}$ that* lim-*generates the function p.*

Proof. Since $\{(r, w) \in \mathbb{Q} \times \Sigma^* \mid r < p(w)\}$ is Π_2^0, there is a TM $\mathcal{M}_<$ such that $\mathcal{M}_<(r, w)$ emits infinitely many signals if, and only if, $r < p(w)$.

We now aim to extend this computation to real numbers. Suppose we have access to an increasing sequence of rationals r_i converging to a real number r. We run $\mathcal{M}_<$ sequentially on inputs (r_i, w), starting $\mathcal{M}_<(r_n, w)$ after each computation $\mathcal{M}_<(r_i, w)$ (with $i < n$) has emitted $n - i$ signals.

Suppose $r < p(w)$. Then $r_i < p(w)$ for all i, so $\mathcal{M}_<(r_i, w)$ emits infinitely many signals for all i. Conversely, if $r > p(w)$, then $r_i \geq p(w)$ for some i, and $\mathcal{M}_<(r_i, w)$ emits only finitely many signals before looping indefinitely.

We can therefore construct a TM $\mathcal{M}'_<$ which, given $w \in \Sigma^*$ and n digits of r, emits infinitely many signals if $r < p(w)$, and finitely many signals if $r > p(w)$

and n is sufficiently large. Similarly, we can construct a TM $\mathcal{M}'_>$ which, given n elements of a decreasing sequence converging to r, emits infinitely many signals if $r > p(w)$, and finitely many signals if $r < p(w)$ and n is sufficiently large.

We now construct an NTM $\mathcal{N}$ to lim-generate the function p. We use a RandomNumber gadget to generate the binary approximants r_i to a random variable r uniformly distributed in $[0,1]$. Notice that the distribution of the r_i is independent of the noise ε. We use $\mathcal{M}'_<$ and $\mathcal{M}'_>$ to compute $r < p(w)$ and $r > p(w)$. The computation is halted by a Timer(c) with $1/2 < c < 2/3$ to ensure that RandomNumber generates sufficiently many bits, but that the Memory is still error-free with high probability.

Fix $w \in \Sigma^*$ and n, and suppose $p(w) \notin [r_n, r_n + 1/2^n]$, which occurs with probability $1 - 1/2^n$. We claim that after a fixed time T, independent of r, the value of the register does not change. In the case $r_n + 1/2^n < p(w)$, then $r_i < p(w)$ for all i, so $\mathcal{M}'_<$ emits infinitely many signals, whereas $\mathcal{M}_>(r_n, w)$ emits only finitely many signals, so after some time $T(r_n)$, machine $\mathcal{M}_>(r_n, w)$ does not emit further signals. Since there are only finitely many possible values of r_n, we can choose T independently of r_n The case $r_n > p(w)$ is similar.

Using the same argument as in Theorem 6, we see that with high probability, $\mathcal{N}_\varepsilon$ accepts if $r_n < p(w)$ and rejects if $p(w) < r_n + 1/2^n$. Hence $\mathbb{P}(\mathcal{N}_\varepsilon(w) = \top) \in [r_n, r_n + 1/2^n]$, and since n is arbitrary, $\mathbb{P}(\mathcal{N}_\varepsilon(w) = \top) = p(w)$. □

To prove that Theorem 8 gives a precise characterisation of the computational power of a converging NTM, we analyse the limit as $\varepsilon \to 0$.

Theorem 9. $\lim_{\varepsilon \to 0} \mathbb{P}(\mathcal{M}_\varepsilon(w) = \top)$ *is* $0'$*-computable for any converging NTM.*

Proof. The function $f(\varepsilon, w)$ given by $\mathbb{P}(\mathcal{M}_\varepsilon(w) = \top)$ is computable when ε is rational, and converges as $\varepsilon \to 0$ for all $w \in \Sigma^*$. By definition,

$$r < \lim_{\varepsilon \to 0} f(\varepsilon, w) \iff \exists s > r,\ \exists q > 0,\ \forall \varepsilon < q,\ (\neg(s > f(\varepsilon, w))).$$

The inequality $(s > f(\varepsilon, w))$ belongs to the class Σ^0_1. We deduce that the relation $(r < \lim_{\varepsilon \to 0} f(\varepsilon, w))$ belongs to the class Σ^0_2, and hence is $0'$-recursively enumerable. Symmetrically, the inequality $(r > \lim_{\varepsilon \to 0} f(\varepsilon, w))$ is also $0'$-recursively enumerable. Hence $\lim_{\varepsilon \to 0} f(\varepsilon, w)$ is $0'$-computable. □

6.2 Generating Upper $0'$-Semicomputable Probabilities

Recall that a function p is upper $0'$-semicomputable if the set $\{(r, w) \in \mathbb{Q} \times \Sigma^* \mid r > p(w)\}$ is of class Σ^0_2, so that $\{(r, w) \in \mathbb{Q} \times \Sigma^* \mid r \leq p(w)\}$ is of class Π^0_2.

Theorem 10. *Let* $p : \Sigma^* \to [0,1]$ *be an upper* $0'$*-semicomputable function. There exists an NTM* $\mathcal{S}$ *that* lim sup*-generates* p.

Proof. We use the machine $\mathcal{M}'_<(r, w)$ and the RandomNumber from the proof of Theorem 8, and combine this with Delay as in the proof of Theorem 7, and a Timer(c) to halt the computation.

Computation proceeds by running $\mathcal{M}'_<$, and starting a DELAY whenever $\mathcal{M}_<$ emits a signal. If $\mathcal{M}_<$ emits signals infinitely often, then every $r_n < p(w)$, so $r \leq p(w)$, and if $\mathcal{M}_<(r_n, w)$ loops for some r_n, then $r_n \geq p(w)$, so $r \geq p(w)$.

The rest of the proof follows that of Theorems 7 and 8, and is omitted. □

Theorem 11. $\limsup_{\varepsilon\to 0} \mathbb{P}\left(\mathcal{M}_\varepsilon(w) = \top\right)$ *is upper* $0'$*-semicomputable.*

Proof. As in the proof of Theorem 9, it is easy to show that $\limsup_{\varepsilon\to 0} f(\varepsilon, w)$ is upper $0'$-semicomputable whenever $f : (0, 1) \times \Sigma^* \to [0; 1]$ is continuous and computable. □

7 Concluding Remarks

We have described a class of randomly perturbed Turing machines and studied their computational properties. We have shown that in the limit of infinitesimal noise, these machines can be programmed to lim-decide Δ^0_2, and lim sup-decide Π^0_2 sets. It is interesting to compare this result with [5], where a small nondeterministic noise led to a Π^0_1 computational power only. We have also given a characterisation of the acceptance probability distributions which can be generated. As a future work we are planning to explore how sensitive are these results to the choice of a computational model (discrete, hybrid or analog) and of a stochastic noise model.

References

1. McKee, W., Bergman, D., Nguyen, N., Aton, T., Block, L., Huynh, V., McAdams, H., Smith, E., McPherson, J., Janzen, J., Ondrusek, J., Hyslop, A., Russell, D., Coy, R.: Cosmic ray neutron induced upsets as a major contributor to the soft error rate of current and future generation DRAMs. In: Proceedings of the International Reliability Physics Symposium. (1996)
2. Howard, J.J., Hardage, D.: Spacecraft environments interactions: Space radiation and its effects on electronic systems. Technical Report TP-1999-209373, NASA (1999) `http://trs.nis.nasa.gov/archive/00000502/01/tp209373.pdf`.
3. Puri, A.: Dynamical properties of timed automata. Discrete Event Dynamic Systems **10** (2000) 87–113
4. Fränzle, M.: Analysis of hybrid systems: An ounce of realism can save an infinity of states. In Flum, J., Rodríguez-Artalejo, M., eds.: Computer Science Logic (CSL'99). Volume 1683 of LNCS., Springer-Verlag (1999) 126–140
5. Asarin, E., Bouajjani, A.: Perturbed Turing machines and hybrid systems. In: Proceedings of the 16th Annual IEEE Symposium on Logic in Computer Science, IEEE Computer Society (2001) 269
6. Delyon, B., Maler, O.: On the effects of noise and speed on computations. Theoretical Computer Science **129** (1994) 279–291
7. Maass, W., Orponen, P.: On the effect of analog noise in discrete-time analog computations. Neural Computation **10** (1998) 1071–1095
8. Freidlin, M., Wentzell, A.: Random perturbations of dynamical systems. Springer-Verlag, New York (1984)

9. Rogers, H.: Theory of Recursive Functions and Effective Computability. McGraw-Hill (1967)
10. Weihrauch, K.: Computable analysis. Springer-Verlag, Berlin (2000)