Theory of Two-sided Experiments
A new insight on measurable numbers
(Extended Abstract)

Tânia Filipa Nascimento Ambaram
Instituto Superior Técnico - Universidde de Lisboa

July, 2014

Abstract

In this work we proceed with the investigation of the power of an abstract model of computation that considers Turing machines coupled with physical oracles. A physical oracle is an experiment controlled by the Turing machine with the purpose of measuring some quantity or parameter, possibly suitable to boost the power of the Turing machine.

We consider one of the three known types of measurement experiments — the two-sided case. Researches on the computational power under bounded resources of the two other types — threshold and vanishing — have been considered and are settle by other authors up to some hard open problems. Herein we address the unfinished case of the two-sided oracle, establishing the corresponding upper bounds.

We consider three types of communication between the Turing machine and the oracle (infinite precision, unbounded precision and fixed precision), that simulate the error propagation common to physical experiments. These three types of precision introduce variants of the two-sided oracle machines. We fix the upper bounds for the three cases.

In the context of the infinite precision protocol of two-sided machines, we then address the question of knowing if a number, a physical parameter, is measurable or not. This problem was first put out by the physicists Geroch and Hartle without a formal framework to reason upon definition. We characterize the measurable and the non measurable numbers and we analyze some of their properties.

1 Paradigm

Although the Turing machine model is the standard model of computation, we cannot deny that other models or extensions of the Turing machine can compute behind the Turing limit. The study of other models of computation have been pursued over the past decades (see [10]). All of them associate with the Turing machine some resources, as infinite memory, non recursive information, infinite specification or infinite computation, in order to improve its computational power.

One of the well studied hypermachines is the oracle Turing machine. This hypermachine consists in a Turing machine coupled with an oracle, which is a set of words. The Turing machine is allowed to question about the set membership of some words to the oracle.

Besides the study of mathematical hypermachine models of computation, other models, based on physical theories, have been introduced, e.g., the neural net model first presented in 1993 by *Hava Siegelmann* (see [11]) as a computation device similar to the Turing machine model, with or without oracle.

A physical process with particular interest for us was presented in [5] by *Beggs and Tucker*. They define a physical experiment, the scatter experiment, capable of approximating any real number up to any precision. From this experiment, and thinking in the oracle Turing machine model, emerged the paradigm of coupling Turing machines with physical experiments. In this paradigm hypercomputation, not being a program, is performed by an abstract physical device (see [9]).

The paradigm is such that the Turing machine is coupled with a physical experiment, called the physical oracle. The physical oracle allows the Turing machine to measure some quantity or parameter. The communication between the Turing machine and the physical oracle is made by the query tape, where the Turing machine writes the parameters needed to initialize the experiment. The consultation of the physical oracle will cost more resources as time than the consultation of the standard oracle, since we need to execute a physical experiment with intrinsic time costs. The communication between the Turing machine and the physical oracle is mediated by protocols (infinite precision, unbounded precision and fixed precision) that define how the data/ information, that is exchanged between the Turing machine and the oracle, is processed and interpreted.

2 Smooth scatter experiment

We consider the Turing machine coupled with a particular physical experiment, the smooth scatter experiment, $SmSE$ (see Figure 1). The use of the smooth scatter experiment is not mandatory since all the two-sided experiments would lead to the same results.

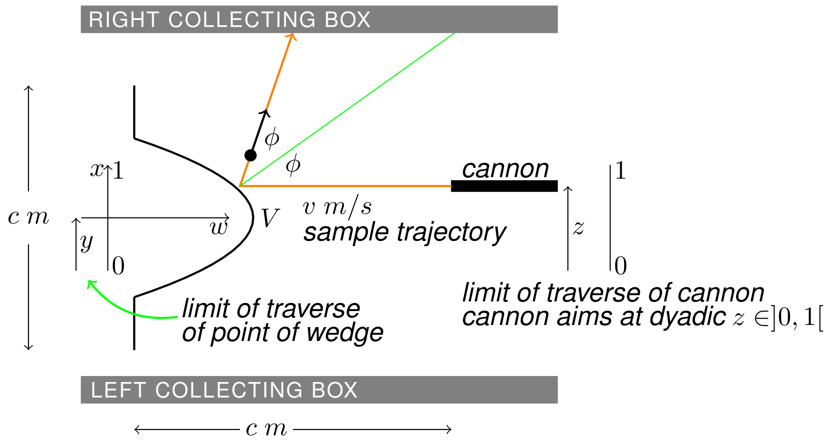

Figure 1: The smooth scatter experiment.

The $SmSE$ is a variation of the sharp scatter experiment (see [1] and [2]), where the sharp wedge is replaced by a smooth wedge given by a function $g(x) \in C^n$, satisfying the following properties:

- $g(x)$ must be convex, i.e., $\frac{d^2}{dx^2} g(x) < 0$;
- $g(x)$ is symmetric with respect to the line of the vertex y, i.e., $g(y - \xi) = g(y + \xi)$;
- $g(x)$ has a maximum on the vertex of the wedge, i.e., $(\frac{d}{dx} g(x))(y) = 0$.

As the sharp scatter experiment, the $SmSE$ measures the position of the vertex y in the smooth wedge. In order to measure the vertex position, the smooth scatter experiment sets a cannon at some position z, shoots a particle from the cannon, and then waits some time until the particle is captured in a detection box. We consider that y can take any value in $]0, 1[$. After shooting, the experiment can have one of the behaviors explained in Table 1.

By analyzing in which box the particle is detected, it is possible to conclude the relative positions of y and z, i.e., whether the vertex is in the right or left side of the cannon. Then, repeating this procedure, by resetting the cannon position and executing one more run of the experiment, we can better approximate the vertex position.

Data: The position z
if $z > y$ **then**
    The particle is detected in the right box;
end
if $z < y$ **then**
    The particle is detected in the left box;
else
    The particle is not detected in any of the two boxes;
end

Table 1: The smooth scatter.

The $SmSE$ is a symmetric two-sided experiment and it is governed by a fragment of Newtonian mechanics. It belongs to the class of physical oracles with physical time (the time intrinsic to the experiment) in $ExpLin$, where:

$$ExpLin = \{g : \mathbb{N} \to \mathbb{N} \mid 2^{an+b} \leq g(n) \leq 2^{cn+d}\}.$$

As all the physical oracles with physical time in ExpLin, the time taken by the $SmSE$ to compare the test value z with the real unknown value y, called the physical time in z, is given by the expression:

$$\frac{A}{|y-z|^q} \leq t(z) \leq \frac{B}{|y-z|^n}, \tag{1}$$

for some rationals $A, B > 0$ and integers $n, q \geq 1$. We consider that n and q are both equal to 1 since the results are essentially the same for other values of n and q.

The set of oracles with physical time in $ExpLin$, called $\mathbb{O}(ExpLin)$, has an important role in this paradigm of computation. The investigations done on some of those physical oracles led to a conjecture, called the *BCT conjecture*. This conjecture, already discussed in [4], states that for all "reasonable" physical theories and for all physical experiments based on them, the physical time intrinsic to the experiment is at least exponential, i.e., the time needed to establish the n-th bit of the parameter that we are measuring is at least exponential in n.

Assuming that BCT is true, all physical oracles are in $\mathbb{O}(ExpLin)$. Thus if this conjecture is true, as it seems to be by evidences, the theoretical project of finding physical systems that allow to measure some physical quantity ever more accurately and ever more efficiently is condemned to failure.

3 Smooth scatter machine

A Turing machine coupled with a smooth scatter experiment is called a smooth scatter machine, $SmSM$. This machine was already studied in [7]. The $SmSM$ uses the oracle to read the vertex position and thus gain information for its computations. Since the physical time of the $SmSE$ goes to infinity when $|y - z|$ goes to zero (see Equation 1), and we consider Turing machines clocked in polynomial time, we must guarantee that the Turing machine does not wait an infinite amount of time for the oracle answer. To do so, we consider that the Turing machine, besides its internal clock, also clocks the consultation of the oracle with some time schedule T, where $T : \mathbb{N} \to \mathbb{N}$ is a time constructible function on the size of the query that returns the waiting time for the oracle answer.

After each consultation of the oracle, if the oracle detects the particle in the right box, the Turing machine will be in the state q_r; if the oracle detects the particle in the left box, the Turing machine will be in the state q_l; and if the oracle does not detect the particle in any box, which means that occurs timeout, the Turing machine will be in the state q_t. Then the machine resumes its computation.

We consider that the queries written by the Turing machine are written in binary and correspond to a dyadic rational, i.e., the position is a number of the form $n/2^k$ where n is an integer in $]0, 2^k[$ and k is a positive integer. This restriction is considered since the query must be finite and the dyadic rationals have a finite binary expansion followed by an infinite sequence of zeros. Thus we consider only dyadic queries which are the finite part of the respective dyadic rational.

Therefore, since we consider that the vertex position is any real number $y \in]0, 1[$, our machine writes in the query tape a word z and the experiment should process the query tape as a dyadic rational, in other words, the experiment should obtain a value z' such that:

$$z' = \sum_{i=1}^{|z|} 2^{-i} z[i], \tag{2}$$

where $z[i]$ represents the i-th bit of z. After processing z, the experiment should set the position of the cannon and execute a run.

The communication between the Turing machine and the $SmSE$ is mediated by protocols, describing how the queries may be interpreted by the experiment. These protocols are considered in order to simulate the experimental errors in the physical experiments. We consider three types of protocols described below:

Protocol 1. *Given a dyadic rational z in the query tape, the experiment sets the position of the cannon to $z' = z$. In this case we are working with an error-free $SmSE$ or $SmSE$ with infinite precision, and with the protocol $Prot\_IP(z)$. A Turing machine coupled with this oracle is called an* error-free smooth scatter machine*;*

Protocol 2. *Given a dyadic rational z in the query tape, the experiment sets the position of the cannon to $z' \in [z - 2^{-|z|}, z + 2^{-|z|}]$, chosen uniformly. In this case we are working with an error-prone $SmSE$ with unbounded precision, and with the protocol $Prot\_UP(z)$. A Turing machine coupled with this oracle is called an* error-prone smooth scatter machine with unbounded precision*;*

Protocol 3. *Given a dyadic rational z in the query tape, the experiment sets the position of the cannon to $z' \in [z - \epsilon, z + \epsilon]$, chosen uniformly, for a fixed $\epsilon = 2^{-q}$, for some positive integer q. In this case we are working with an error-prone $SmSE$ with fixed precision, and with the protocol $Prot\_FP(z)$. A Turing machine coupled with this oracle is called an* error-prone smooth scatter machine with fixed precision*.*

Although in this work we consider ϵ such that $\epsilon = 2^{-q}$ for some positive integer q, the same computational results are obtained if we consider any real ϵ.

Consider that $m\lfloor_\ell$ denotes the first ℓ digits of m, if m has ℓ or more than ℓ digits, otherwise it represents m padded with k zeros, for some k, until we get $|m0^k| = \ell$. The pruning or the padding technique is used in order to control the time schedule during the measurement process.

For each communication protocol we use different measurement algorithms (see Algorithms 1, 2 and 3), as we must guarantee that we can approximate the vertex position up to any precision independently of the communication protocol. The measurement algorithm for the fixed precision is different from the other two as it measures a relative frequency.

4 Computational power

The $SmSM$ extend the computational power of the Turing machines to nonuniform complexity classes. Note that the computational results that are valid for the $SmSM$ are also valid for any Turing machine coupled with a

Algorithm 1: Measurement algorithm for infinite precision.

Data: Positive integer ℓ representing the desired precision
$x_0 = 0$, $x_1 = 1$, $z = 0$;
while $x_1 - x_0 > 2^{-\ell}$ **do**
  $z = (x_0 + x_1)/2$;
  $s = Prot\_IP(z\lfloor_\ell)$;
  if $s ==$ "q_r" **then**
    $x_1 = z$;
  end
  if $s ==$ "q_l" **then**
    $x_0 = z$;
  else
    $x_0 = z$; $x_1 = z$;
  end
end
return *dyadic rational denoted by* x_0

Algorithm 2: Measurement algorithm for unbounded precision.

Data: Positive integer ℓ representing the precision
$x_0 = 0$, $x_1 = 1$, $z = 0$;
while $x_1 - x_0 > 2^{-\ell}$ **do**
  $m = (x_0 + x_1)/2$;
  $s = Prot\_UP(z\lfloor_\ell)$;
  if $s ==$ "q_r" **then**
    $x_1 = z$;
  end
  if $s ==$ "q_l" **then**
    $x_0 = z$;
  else
    $x_0 = z$; $x_1 = z$;
  end
end
return *dyadic rational denoted by* x_0

Algorithm 3: Measurement algorithm for fixed precision.

Data: Integer ℓ representing the precision
$c = 0$, $i = 0$, $\xi = 2^{2\ell+h}$;
while $i < \xi$ **do**
  $s = Prot\_FP(1\lfloor_\ell)$;
  if $s ==$ "q_l" **then**
    $c = c + 2$;
  end
  if $s ==$ "q_t" **then**
    $c = c + 1$;
  end
  i++;
end
return $c/(2\xi)$

physical oracle with physical time in $ExpLin$. Given the measurement algorithms, for each communication protocol, we were interested in clarifying the lower bound complexity results and fix the upper bound complexity results.

In the lower bounds the goal is, given any set belonging to some suitable nonuniform complexity class, we will specify a $SmSM$ that decides that set. This result is obtained by choosing the $SmSE$ that codifies in its vertex position the information that we need to decide the set and also by using the $SmSE$ as a generator of random sequences. As we need to read the the vertex position, we need to consider a bound for the physical time. In this case the bound is $C/\ |y - z|$, for some $C > 0$, which is exponential in the precision of the query. Note that, since for all two-sided oracles with physical time in $ExpLin$ we can define a bound for the physical time, the computational results are the same for all of them. We clarify the following results, already proved in [7]:

Theorem 1. *If $B \in P/log*$, then there exists an error-free $SmSM$, clocked in polynomial time, that decides B.*

Theorem 2. *If $B \in BPP//log*$, then there exists an error-prone $SmSM$ with unbounded precision (with fixed precision $\epsilon = 2^{-q}$, for some positive integer q), clocked in polynomial time, that decides B.*

Our goal with this dissertation was to fix some suitable nonuniform class that includes all sets decidable by $SmSM$ with real parameters. We constructed advice functions to codify for the information needed to simulate $SmSE$ queries. We found that logarithmic squared advices suffice to codify the approximations to the so-called boundary numbers as in [3]. After words, as in [3], we used the explicit time technique to reduce logarithmic squared advice to logarithmic advice. Since there are always boundary numbers, for the two-sided oracles with physical time in $ExpLin$, we conclude that the upper bounds results are common to all two-sided physical oracles with physical time in $ExpLin$. We got the following results:

Theorem 3. *If B is decidable by a $SmSM$ with infinite precision and exponential protocol, clocked in polynomial time, then $B \in P/log^2*$.*

Theorem 4. *If B is decidable by a $SmSM$ with infinite precision and exponential protocol $T(k) \in \Omega(2^{k/2})$, clocked in polynomial time, then $B \in P/log*$.*

Theorem 5. *If B is decidable by a $SmSM$ with unbounded precision (with fixed precision $\epsilon = 2^{-q}$, for some positive integer q) and exponential protocol T, clocked in polynomial time, then $B \in BPP//log^2*$.*

Theorem 6. *If B is decidable by a $SmSM$ with unbounded precision (with fixed precision $\epsilon = 2^{-q}$, for some positive integer q) and exponential protocol T, clocked in polynomial time, then, considering explicit time, $B \in BPP//log*$.*

5 Measurable numbers

In the paradigm of Turing machine coupled with physical oracle, the capability of computing behind the Turing limit is obtained with an abstract physical experiment that allows the Turing machine to measure some physical quantity and use it during its computations. Therefore the numbers that we can measure using the physical oracle are essential to extend the computational power of the Turing machine to nonuniform complexity classes. Hence we are interested in the characteristics of these numbers and also the characteristics of the non measurable numbers. The measurability results were proved for any physical oracle with physical time in $ExpLin$. Let such oracle, with attribute y, be denoted by $\mathcal{A}(y)$.

The discussion about measurable numbers was first introduced in [8]. In this article *Geroch* and *Hartle* proposed in parallel with the notion of computable number, the notion of measurable number. They also argued about the existence of measurable numbers that are not computable and vice versa. Other works were done, as [4] and [6], as an effort to formalize the concept of measurable number.

Generally, a measurable number w is a number for which there exists a physical experiment of measuring w, that for example has w as a parameter in its experimental apparatus, such that a Turing machine coupled with that physical experiment obtained approximations of w up to any precision, by performing the experiment. More formally we have:

Definition 1. A value y is said to be measurable if we can find a measurement algorithm running on a Turing machine M coupled with $\mathcal{A}(y)$ such that, given a time constructible schedule T, there exists $p \in \mathbb{N}$ such that, for all $n \geq p$, M on input n outputs a dyadic rational r, such that $|y - r| < 2^{-n+1}$, without having timeout on any query.

Based on this definition we characterized the set of measurable numbers y as the non dyadic real values $y \in]0, 1[$ given by the pattern:

$$y = 0.\underbrace{1...1}_{u_1}\underbrace{0...0}_{u_2}\underbrace{1...1}_{u_3}\underbrace{0...0}_{u_4}\underbrace{1...1}_{u_5}\underbrace{0...0}_{u_6}..., \tag{3}$$

where each u_k gives the number of bits in the k-th group, for $k \in \mathbb{N}$; $u_1 \geq 0$; and $u_i \geq 1$ for $i \geq 2$; and the sequences u_k are such that they are bounded by a computable function f.

The set of non measurable numbers y is described as the dyadic rationals plus the non dyadic real values $y \in]0, 1[$ given by the Pattern 3, but where the sequences u_k are not bounded by any computable function f. Note that dyadic rationals cannot be given by the above pattern, since from certain order they only have 0 bits. Therefore, the uncontrolled sequences are responsible for the non measurability property.

We recall the results proved in [6] that show that although uncountable many values y are non measurable, almost all the possible unknown values are measurable in the sense of Measurement Theory. Since the communication between the Turing machine and the physical experiment is mediated by protocols and by some time schedule, we also considered a definition of measurable number where the schedule is fixed.

Definition 2. A value y is said to be measurable with time constructible schedule $T \in \mathbb{C}$, if we can find a measurement algorithm running on a Turing machine M coupled with $\mathcal{A}(y)$ such that, there exists $p \in \mathbb{N}$ such that, for all $n \geq p$, M on input n outputs a dyadic rational r, such that $|y - r| < 2^{-n+1}$, without having timeout on any query.

Based on this definition we proved the following theorem:

Theorem 7. *If T_1 and T_2 are two schedules such that $T_1 \ll T_2$, then the set of measurable numbers relative to schedule T_1 coincide with the set of measurable numbers relative to schedule T_2 (i.e., absolute measurement does not depend on the growing rate of the schedule).*[1]

6 Properties of measurable numbers

We investigated if (non) measurability is preserved under arithmetical operations. E.g., computable numbers are closed under arithmetical operations, i.e., if w and s are computable numbers, then also are $w + s$, $w - s$, ws and w/s for $s \neq 0$.

We worked on ω-sequences of bits, i.e., infinite sequences x over the binary alphabet. This means that $x : \mathbb{N} \to \{0, 1\}$. Thus we start by clarifying the meaning of an arithmetical operation on ω-sequences. Recall that we consider that all the values $y \in]0, 1[$ do not contain infinite sequences of 1's. Therefore we consider the following general definition of arithmetical operation:

Definition 3. Let w and s be real values in $]0, 1[$. We define $w * s$, for $* \in \{+, -, \times, \div\}$, as

$$\lim_{n\to\infty} w\rfloor_n * s\rfloor_n \ .$$

[1] Let V and T be two time schedule functions. We say that $V \ll T$ if $V \in o(T)$.

Whenever such operations occur, the result is supposed to be in the limit case.

To study the closure under addition and subtraction we first look at the properties of adding and subtracting sequences of bits, considering the case where there is no bit being shifted from the sequences previously summed or subtracted, and the opposite case. Based on those properties and thinking about how to annihilate or generate sequences we discovered counterexamples that allow us to conclude the non closure of non measurable and measurable numbers under addition and subtraction (see Tables 2, 3, 4 and 5).

00 ... 00 $+00$ ... 00 00 ... 00	00 ... 00 $+11$ ... 11 11 ... 11
11 ... 11 $+00$ ... 00 11 ... 11	11 ... 11 $+11$ ... 11 Shift 1 $\leftarrow 11$ ... 10

Table 2: Sum of sequences of bits.

00 ... 00 $+00$ ... 00 00 ... $01\leftarrow$ Shift 1	00 ... 00 $+11$ ... 11 Shift 1 $\leftarrow 00$ ... $00\leftarrow$ Shift 1
11 ... 11 $+00$ ... 00 Shift 1 $\leftarrow 00$ ... $00\leftarrow$ Shift 1	11 ... 11 $+11$ ... 11 Shift 1 $\leftarrow 11$ ... $11\leftarrow$ Shift 1

Table 3: Sum of sequences of bits with a bit being shifted.

We concluded the following results:

Theorem 8. *The set of non measurable numbers is not closed under addition and subtraction.*

Theorem 9. *The set of measurable numbers is not closed under addition and subtraction.*

For the non measurable numbers, just looking at the properties of the dyadic rationals, we also conclude that the set of non measurable numbers is not closed under division.

Theorem 10. *The set of non measurable numbers is not closed under division.*

Consider that $a_k = u_1 + u_2 + ... + u_k$ for all $k \geq 0$, i.e. a_k is the k-th position where the block of bits of a number given by the Pattern 3 change.

$00 \ldots 00$ $-00 \ldots 00$ $00 \ldots 00$	$00 \ldots 00$ $-11 \ldots 11$ Shift 1 $\leftarrow 00 \ldots 01$
$11 \ldots 11$ $-00 \ldots 00$ $11 \ldots 11$	$11 \ldots 11$ $-11 \ldots 11$ $00 \ldots 00$

Table 4: Subtraction of sequences of bits.

$00 \ldots 00$ $-00 \ldots 00$ Shift 1 $\leftarrow 11 \ldots 11 \leftarrow$ Shift 1	$00 \ldots 00$ $-11 \ldots 11$ Shift 1 $\leftarrow 00 \ldots 00 \leftarrow$ Shift 1
$11 \ldots 11$ $-00 \ldots 00$ $11 \ldots 10 \leftarrow$ Shift 1	$11 \ldots 11$ $-11 \ldots 11$ Shift 1 $\leftarrow 00 \ldots 00 \leftarrow$ Shift 1

Table 5: Subtraction of sequences of bits with a bit being shifted.

For the measurable numbers we conclude that the non closure under addition or subtraction could be derived from the fact that a_k is not computable. Thus we considered the subclass of measurable numbers, where all the elements have computable structure.

Definition 4. We say that a number w given by the Pattern 3 has computable structure, if u_k is a computable function for all $k \in \mathbb{N}$.

We conclude that this subset of measurable numbers numbers coincide with the non dyadic computable numbers in $]0,1[$.

Theorem 11. *Let w be a number given by the Pattern 3. Then w has computable structure if and only if w is computable.*

Then we conclude the immediate result noticing that given any two measurable numbers with computable structure, they are computable numbers, and thus the result of these numbers under any arithmetic operation is also a computable number, which has computable structure and thus it is measurable.

Theorem 12. *Let w and s be two measurable numbers with computable structure, such that $w, s \in]0,1[$. Then $w * s$ is measurable, for $* \in \{+, -, \times, \div\}$, unless $w * s$ is a dyadic rational.*

7 Open Problems

For the two-sided experiments with physical time in $ExpLin$ remains the question if for the infinite precision case the lower and the upper bounds may coincide without doing assumptions on the schedule. For the two-sided experiments with non infinite precision remains the question if it is possible to make coincide the lower and the upper bounds without using the explicit time technique. Namely, it is an open problem if it is possible to show that either the upper bound falls to $BPP//log*$ or that there exists a set decidable in polynomial time by a $SmSM$ with non infinite precision that is not in $BPP//log*$.

In the matter of measurability, stands the question if the non measurable numbers are or not closed under multiplication and if the measurable numbers without computable structure are closed under multiplication and division. As multiplication and division are operations derived from addition and subtraction, respectively, probably the non closure for these two operations is immediate. We believe that the same techniques used in addition and subtraction operations could be used to prove the closure or the non closure of the (non) measurable numbers under multiplication and division, although such operations are more difficult to analyze.

References

[1] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker, *Computational complexity with experiments as oracles*, Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences) **464** (2008), no. 2098, 2777–2801.

[2] Edwin Beggs, José Félix Costa, Bruno Loff, and John V. Tucker, *Computational complexity with experiments as oracles. II. Upper bounds*, Proceedings of the Royal Society, Series A (Mathematical, Physical and Engineering Sciences) **465** (2009), no. 2105, 1453–1465.

[3] Edwin Beggs, José Félix Costa, Diogo Poças, and John V. Tucker, *Oracles that measure thresholds: the Turing machine and the broken balance*, Journal of Logic and Computation **23** (2013), no. 6, 1155–1181.

[4] Edwin Beggs, José Félix Costa, and John V. Tucker, *Computational models of measurement and hempel's axiomatization*, Causality, Meaningful Complexity and Embodied Cognition, Theory and Decision Library A, Springer, 2009, pp. 155–184.

[5] Edwin Beggs and John V. Tucker, *Experimental computation of real numbers by newtonian machines*, Proceedings of the Royal Society, Series A **463** (2007), no. 2082, 1541–1561.

[6] Edwin J. Beggs, José Félix Costa, and J. V. Tucker, *Limits to measurement in experiments governed by algorithms*, Mathematical Structures in Computer Science **20** (2010), no. 06, 1019–1050.

[7] Edwin J. Beggs, José Félix Costa, and J. V. Tucker, *The impact of models of a physical oracle on computational power*, Mathematical Structures in Computer Science **22** (2012), no. 05, 853–879.

[8] Robert Geroch and James B. Hartle, *Computability and physical theories*, Foundations of Physics **16** (1986), no. 6, 533–550.

[9] Bruno Loff and José Félix Costa, *Five Views of Hypercomputation.*, IJUC **5** (2009), no. 3-4, 193–207.

[10] Toby Ord, *Hypercomputation: computing more than the Turing machine*, Master's thesis, The University of Melbourne, 2002.

[11] Hava T. Siegelmann and Eduardo D. Sontag, *Analog Computation via Neural Networks*, Theoretical Computer Science. **131** (1993), no. 2, 331–360.