Asymptotic Logical Uncertainty and The Benford Test

Scott Garrabrant[1,2], Siddharth Bhaskar[1], Abram Demski[2,3], Joanna Garrabrant, George Koleszarik, and **Evan Lloyd[1]**
[1]University of California, Los Angeles
[2]Machine Intelligence Research Institute
[3]University of Southern California

Abstract

We give an algorithm $A_{L,T}$ which assigns probabilities to logical sentences. For any simple infinite sequence $\{\phi_{s_n}\}$ of sentences whose truth-values appear indistinguishable from a biased coin that outputs "true" with probability p, we have $\lim_{n\to\infty} A_{L,T}(s_n) = p$.

1 Introduction

Let $\phi_1, \phi_2, \ldots$ be a simple enumeration of all sentences in first order logic over ZFC. The goal of logical uncertainty is to construct an algorithm M which on input N outputs a probability $M(N)$, which represents the probability that ϕ_N is true [1, 2, 3, 4].[1] This notion of probability does not refer to random variables. It refers to the degree of uncertainty that one might have about logical sentences whose truth-values have not been calculated.

Much work has been done on a related problem where M on input N outputs an infinite sequence of numbers and $M(N)$ is defined to be the limit of the sequence output by M on input N [1, 2, 11]. In this case, $M(N)$ is not computable, and can easily be 1 for all provable ϕ and 0 for all disprovable ϕ, so all of the work is in figuring out how M should behave when ϕ is independent of ZFC.

In this paper, we take a different approach, which we call *asymptotic logical uncertainty*. We require that $M(N)$ be computable and have runtime bounded by some function of N.

We propose as a baseline that any method of quickly assigning probabilities should be able to pass a test we call the *Benford test*. Consider the infinite sequence of sentences $\{\phi_{s_n}\}$ given by ϕ_{s_n} = "The first digit of $3 \uparrow^n 3$ is a 1." We say that M passes the Benford test if

$$\lim_{n\to\infty} M(s_n) = \log_{10}(2) \approx .30103,$$

as prescribed by Benford's law. More generally, we say that M passes the generalized Benford test if it converges to the correct probability on any similar infinite sequences whose truth values appear indistinguishable from independent flips of a biased coin. We then give an algorithm $A_{L,T}$ which passes the generalized Benford test.

Logical uncertainty is one aspect of the problem of combining probability and logic, of which *statistical relational learning* is another [12]. Statistical relational learning addresses the problem of representing probabilistic models with logical structure, including regularities such as repeated entities and other complexities such as uncertainty about the number of entities. In contrast, logical uncertainty deals with uncertainty *about* logic. As Paul Christiano put it: "any realistic agent is necessarily uncertain not only about its environment or about the future, but also about the logically necessary consequences of its beliefs." [1]

2 The Benford Test

Benford's law states that in naturally occurring numbers, the leading digit $d \in \{1, \ldots, 9\}$ of that number in base 10 occurs with probability $\log_{10}(1 + \frac{1}{d})$. Many mathematical sequences have been shown to have frequencies of first digits that satisfy Benford's law [13]. In particular, the frequencies of the first digits of powers of 3 provably satisfy Benford's law.

The function $3 \uparrow^n k$ is defined by $3 \uparrow^1 k = 3^k$, $3 \uparrow^n 1 = 3$, and $3 \uparrow^n k = 3 \uparrow^{n-1} (3 \uparrow^n (k-1))$. Throughout the paper, let $T(N)$ be an increasing time complexity function in the range of $N \leq T(N) \leq 3 \uparrow^k N$ for some fixed k, and let $R(N) = T(N)N^4 \log T(N)$.

Consider the sequence $3 \uparrow^n 3$. Clearly this sequence only contains powers of 3. We might hypothesize that the frequencies of the first digits in this sequence also satisfy Benford's law. However, $3 \uparrow^n 3$ is very large, and first digit of $3 \uparrow^n 3$ is probably very difficult to

Research supported by the Machine Intelligence Research Institute (intelligence.org). Technical Report 2015–11.

[1]The problem has also been studied in the case where we don't require computability even in the limit [5, 6, 7]. The problem was first studied in the context of measures on Boolean algebras [8, 9, 10].

compute. It is unlikely that the first digit of $3 \uparrow^3 3$ will ever be known.

If asked to quickly assign a probability to the sentence ϕ_{s_n} = "The first digit of $3 \uparrow^n 3$ is a 1," for some $n > 2$, the only reasonable answer would be $\log_{10}(2) \approx .30103$. Note that ϕ_{s_n} is either true or false; there are no random variables. The probability here represents a reasonable guess in the absence of enough time or resources to compute $3 \uparrow^n 3$.

Definition 2.1. *Let M be a Turing machine which on input N runs in time $O(R(N))$ and outputs a probability $M(N)$, which represents the probability assigned to ϕ_N. We say that M passes the Benford test if*

$$\lim_{n\to\infty} M(s_n) = \log_{10}(2),$$

where ϕ_{s_n} = "The first digit of $3 \uparrow^n 3$ is a 1."

It is easy to pass the Benford test by hard-coding in the probability. It is more difficult to pass the Benford test in a natural way. That the best probability to assign to ϕ_{s_n} is $\log_{10}(2)$ depends not only on the fact that the frequency with which ϕ_{s_n} is true tends toward $\log_{10}(2)$, but also on the fact that the sequence of truth-values of ϕ_{s_n} contains no patterns that can be used to quickly compute a better probability on some subsequence. We therefore assume that this sequence of truth-values is indistinguishable from a sequence produced by a coin that outputs "true" with probability $\log_{10}(2)$. Formally, we are assuming that $S = \{s_n | n \in \mathbb{N}\}$ is an *irreducible pattern* with probability $\log_{10}(2)$, as defined in the next section.

3 Irreducible Patterns

Fix a universal Turing machine U and an encoding scheme for machines, and let $U(M, x)$ denote running the machine U to simulate M with input x.

Definition 3.1. [2] *Let $S \subseteq \mathbb{N}$ be an infinite subset of natural numbers such that ϕ_N is provable or disprovable for all $N \in S$, and there exists a Turing machine Z such that $U(Z, N)$ runs in time $T(N)$ and accepts N if and only if $N \in S$.*

We say that S is an irreducible pattern *with probability p if there exists a constant c such that for every positive integer $m \geq 3$ and every Turing machine W expressible in $k(W)$ bits, if*

$$S' = \{N \in S \mid U(W, N) \text{ accepts in time } T(N)\}$$

has at least m elements and $r(m, W)$ is the probability that ϕ_N is provable when N is chosen uniformly at random from the first m elements of S', we have

$$|r(m, W) - p| < \frac{ck(W)\sqrt{\log\log m}}{\sqrt{m}}.$$

The intuition behind the formula is that the observed frequency $r(m, W)$ for any sequence S' we select should not stray far from p. The right hand side of the inequality needs to shrink slowly enough that a true random process would stay within it with probability 1 (given choice of c sufficiently large to accommodate initial variation). The law of the iterated logarithm gives such a formula, which is also tight in the sense that we cannot replace it with a formula which diminishes more quickly as a function of m.

Proposition 3.2. *If we replace provability in Definition 3.1 with a random process, such that for each $N \in S$ the sentence ϕ_N is independently called "provable" with probability p, then S would almost surely be an irreducible pattern with probability p.*

Proof. Fix a Turing machine W. By the law of the iterated logarithm, there exists a constant c_1 such that

$$\limsup_{m\to\infty} \frac{|mr(m, W) - mp|}{\sqrt{m \log\log m}} = c_1$$

almost surely. Therefore

$$\sup_m \frac{|mr(m, W) - mp|}{\sqrt{m \log\log m}} < \infty$$

almost surely. We will use $\Phi(W)$ as a shorthand for this supremum. For any $\varepsilon > 0$, there therefore exists a c_2 such that $\mathbb{P}(\Phi(W) > c_2) \leq \varepsilon$.

We now show that $\mathbb{P}(\Phi(W) > 2c_2 + 1) \leq \varepsilon^2$. By the chain rule for probabilities, it suffices to show that $\mathbb{P}((\Phi(W) > 2c_2 + 1) | (\Phi(W) > c_2)) \leq \varepsilon$. Assume $\Phi(W) > c_2$, and Let m_1 be the first m such that

$$\frac{|mr(m, W) - mp|}{\sqrt{m \log\log m}} > c_2.$$

It suffices to show that the probability that there exists an m_2 with

$$\frac{|m_2 r(m_2, W) - m_2 p|}{\sqrt{m_2 \log\log m_2}} - \frac{|m_1 r(m_1, W) - m_1 p|}{\sqrt{m_1 \log\log m_1}} > c_2$$

is at most ε.

Observe that

$$\frac{|m_2 r(m_2, W) - m_2 p|}{\sqrt{m_2 \log\log m_2}} - \frac{|m_1 r(m_1, W) - m_1 p|}{\sqrt{m_1 \log\log m_1}} \leq \frac{|m_2 r(m_2, W) - m_1 r(m_1, W) - (m_2 - m_1)p|}{\sqrt{(m_2 - m_1)\log\log(m_2 - m_1)}},$$

[2] We tailored this definition of irreducible pattern to our needs. The theory of algorithmic randomness may offer alternatives. However, algorithmic randomness generally considers all computable tests and focuses on the case where $p = \frac{1}{2}$ [14, 15, 16]. We believe that any reasonable definition inspired by algorithmic randomness would imply Definition 3.1.

and that the probability there exists an m_2 with

$$\frac{|m_2 r(m_2, W) - m_1 r(m_1, W) - (m_2 - m_1)p|}{\sqrt{(m_2 - m_1)\log\log(m_2 - m_1)}} > c_2$$

is the same as the probability that $\Phi(W) > c_2$, which is at most ε.

We have thus shown that for every ε, there exists a constant $c_3 = c_2 + 1$ such that the probability that $\Phi(W) \geq 2^\ell c_3$ is at most ε^{2^ℓ}.

Partition the set of all Turing machines into sets $\mathcal{W}_1, \mathcal{W}_2, \ldots$, such that $\mathcal{W}_\ell$ contains all Turing machines expressed in at least 2^ℓ but fewer than $2^{\ell+1}$ bits. The probability that a Turing machine W in $\mathcal{W}_\ell$ violates

$$|r(m, W) - p| < \frac{c_3 k(W)\sqrt{\log\log m}}{\sqrt{m}}, \qquad (\star)$$

for any $m \geq 3$ is at most ε^{2^ℓ}. The number of Turing machines in $\mathcal{W}_\ell$ is at most $2^{2^{\ell+1}}$, so the probability that there is any $W \in \mathcal{W}_\ell$ and $m \geq 3$ which violate $(\star)$ is at most $\varepsilon^{2^\ell} 2^{2^{\ell+1}}$. Therefore, the probability that there is any Turing machine W and $m \geq 3$ which violate $(\star)$ is at most

$$\sum_{\ell \in \mathbb{N}} \varepsilon^{2^\ell} 2^{2^{\ell+1}} = \sum_{\ell \in \mathbb{N}} (4\varepsilon)^{2^\ell}.$$

For small enough ε this goes to 0, so for large enough c_3, the probability that $(\star)$ holds for all W and m goes to 1. Therefore, with probability 1, there exists a c such that

$$|r(m, W) - p| < \frac{ck(W)\sqrt{\log\log m}}{\sqrt{m}},$$

for all m and W. □

We now use the concept of irreducible patterns to generalize the Benford test.

Definition 3.3. *Let M be a Turing machine which on input N runs in time $O(R(N))$ and outputs a probability $M(N)$, which represents the probability assigned to ϕ_N. We say that M passes the generalized Benford test if*

$$\lim_{\substack{N \to \infty \\ N \in S}} M(N) = p,$$

whenever S is an irreducible pattern with probability p.

Note that if we conjecture that the S from Definition 2.1 is an irreducible pattern with probability $\log_{10}(2)$, then any M which passes the generalized Benford test also passes the Benford test.

4 A Learning Algorithm

We now introduce an algorithm $A_{L,T}$ that passes the generalized Benford test (see Algorithm 1).

Let L be the Turing machine which accepts on input N if ZFC proves ϕ_N, rejects on input N if ZFC

Algorithm 1 $A_{L,T}(N)$

$P = 0$
$M = N$
for $j = 0, \ldots, N$ **do**
  $M_Y = 0$
  for Y a Turing machine expressible in $K_Y < \log N$ bits **do**
    $M_X = N$
    for X a Turing machine expressible in $K_X < \log N$ bits **do**
      if $U(X, N)$ and $U(Y, N)$ both accept in time $T(N)$ **then**
        $A = 0$
        $R = 0$
        $i = 1$
        while $i \leq N$ **do**
          if $U(X, i)$ and $U(Y, i)$ both accept in time $T(i)$ **then**
            if $U(L, i)$ accepts in time $T(N)$ **then**
              $A = A + 1$
            else if $U(L, i)$ rejects in time $T(N)$ **then**
              $R = R + 1$
            else
              $i = N$
          $i = i + 1$
        $F = A/(A + R)$
        $Q = A + R$
        if $\max\left(K_X, \frac{|F - \frac{j}{N}|\sqrt{Q}}{K_Y \sqrt{\log\log Q}}\right) < M_X$ **then**
          $M_X = \max\left(K_X, \frac{|F - \frac{j}{N}|\sqrt{Q}}{K_Y \sqrt{\log\log Q}}\right)$
    if $M_X > M_Y$ **then**
      $M_Y = M_X$
  if $M_Y < M$ **then**
    $M = M_Y$
    $P = j/N$
return P

disproves ϕ_N, and otherwise does not halt. For convenience, in Algorithm 1, we define $\log q = 1$ for $q < 2$.

Let $TM(N)$ be the set of all Turing machines X expressible in at most $\log N$ bits such that $U(X, N)$ accepts in time at most $T(N)$. The encoding of Turing machines must be prefix-free, which in particular means that no Turing machine is encoded in 0 bits. Let J_N denote the set of rational numbers of the form $\frac{j}{N}$ with $j = 0, \ldots, N$.

For X and Y Turing machines, let $K(X)$ be the number of bits necessary to encode X. Let $S'(X, Y)$ be the subset of natural numbers i which are accepted by both $U(X, i)$ and $U(Y, i)$ in time at most $T(i)$. Let $Q_N(X, Y)$ be the greatest number less than or equal to N such that for every s in the first $Q_N(X, Y)$ elements of S', $U(L, s)$ halts in time $T(N)$. Let $F_N(X, Y)$ be the proportion of the first $Q_N(X, Y)$ elements of S' which L accepts. Let

$$B_N(X, Y, P) = \max\left(K(X), \frac{|F_N(X, Y) - P|\sqrt{Q_N(X, Y)}}{K(Y)\sqrt{\log\log Q_N(X, Y)}}\right).$$

Lemma 4.1. *The output of $A_{L,T}$ on input N is in*

$$\arg\min_{P \in J_N} \max_{Y \in TM(N)} \min_{X \in TM(N)} B_N(X, Y, P).$$

Proof. The algorithm has three **for** loops, the outer ranging over $j = 0, \ldots N$ and the inner two ranging over Y and X respectively, both restricted to Turing machines expressible in $\log N$ bits. The condition on line 8 means that X and Y effectively range over all Turing machines in $TM(N)$, and $P = \frac{j}{N}$ ranges over J_N.

The inner **while** loop will increment the variables A or R a total of exactly $Q_N(X, Y)$ times. Thus, Q is set to $Q_N(X, Y)$ in line 22. Similarly, F is sent to $F_N(X, Y)$ in line 21. Clearly K_X and K_Y are $K(X)$ and $K(Y)$ respectively. Therefore, the expression on lines 23 and 24 is $B_N(X, Y, P)$.

Considering the **for** loops from inner to outer, we minimize this quantity in X, maximize it in Y, and find P of the form j/N minimizing the whole quantity. The P returned is therefore a minimizer of

$$\max_{Y \in TM(N)} \min_{X \in TM(N)} B_N(X, Y, P).$$

□

The code is not optimized for computational efficiency. The following proposition is just to ensure that the runtime is not far off from $T(N)$.

Proposition 4.2. *The runtime of $A_{L,T}(N)$ is in $O(R(N)) = O(T(N)N^4 \log T(N)))$.*

Proof. Simulating U on any input for T time steps can be done in time $cT \log T$ for some fixed constant c [17]. The bulk of the runtime comes from simulating Turing machines on lines 8, 13, 14, and 16. Each of these lines takes at most $cT(N) \log T(N)$ time, and we enter each of these lines at most N^4 times. Therefore, the program runs in time $O(T(N)N^4 \log T(N))$. □

5 Passing the Generalized Benford Test

We are now ready to show that $A_{L,T}$ passes the generalized Benford test. The proof will use the following two lemmas.

Lemma 5.1. *Let S be an irreducible pattern with probability p, and let Z be a Turing machine such that $U(Z, N)$ accepts in time $T(N)$ if and only if $N \in S$.*

There exists a constant C such that if $N \in S$, then there exists a $P \in J_N$ such that

$$\max_{Y \in TM(N)} B_N(Z, Y, P) < C.$$

Proof. Let $P = \frac{\lfloor pN \rfloor}{N}$. From the definition of irreducible pattern, we have that there exists c such that for all Y,

$$|F_N(Z, Y) - p| < \frac{cK(Y)\sqrt{\log\log Q_N(Z, Y)}}{\sqrt{Q_N(Z, Y)}}.$$

Clearly,

$$|P - p| \leq \frac{1}{N} \leq \frac{1}{Q_N(Z, Y)} \leq \frac{1}{\sqrt{Q_N(Z, Y)}} \leq \frac{K(Z)K(Y)\sqrt{\log\log Q_N(Z, Y)}}{\sqrt{Q_N(Z, Y)}}.$$

Setting $C = K(Z) + c$, we get

$$|F_N(Z, Y) - P| \leq |F_N(Z, Y) - p| + |P - p| < \frac{CK(Y)\sqrt{\log\log Q_N(Z, Y)}}{\sqrt{Q_N(Z, Y)}},$$

so

$$\frac{|F_N(Z, Y) - P|\sqrt{Q_N(Z, Y)}}{K(Y)\sqrt{\log\log Q_N(Z, Y)}} < C.$$

Clearly, $K(Z) < C$, so $B_N(Z, Y, P) > C$ for all Y. Therefore,

$$\max_{Y \in TM(N)} B_N(Z, Y, P) < C.$$

□

Lemma 5.2. *Let S be an irreducible pattern with probability p, and let Z be a Turing machine such that $U(Z, N)$ accepts in time $T(N)$ if and only if $N \in S$.*

For all C, for all $\varepsilon > 0$, for all N sufficiently large, for all $P \in J_N$, if $N \in S$, and

$$\min_{X \in TM(N)} B_N(X, Z, P) < C,$$

then $|P - p| < \varepsilon$.

Proof. Fix a C and a $\varepsilon > 0$. It suffices to show that for all N sufficiently large, if $N \in S$ and $|P - p| \geq \varepsilon$, then for all $X \in TM(N)$, we have $B_N(X, Z, P) \geq C$.

Observe that since $B_N(X, Z, P) \geq K(X)$, this claim trivially holds when $K(X) \geq C$. Therefore we only have to check the claim for the finitely many Turing machines expressible in fewer than C bits.

Fix an arbitrary X. Since S is an irreducible pattern, there exists a c such that

$$|F_N(X, Z) - p| < \frac{cK(Z)\sqrt{\log\log Q_N(X, Z)}}{\sqrt{Q_N(X, Z)}}.$$

We may assume that $S'(X, Z)$ is infinite, since otherwise if we take $N \in S$ large enough, $X \notin TM(N)$. Thus, by taking N sufficiently large, we can get $Q_N(X, Z)$ sufficiently large, and in particular satisfy

$$\frac{\sqrt{Q_N(X, Z)}}{K(Z)\sqrt{\log\log Q_N(X, Z)}}\varepsilon \geq C + c.$$

Take $N \in S$ large enough that this holds for each $X \in TM(N)$ with $K(X) < C$, and assume $|P - p| \geq \varepsilon$. By the triangle inequality, we have

$$\begin{aligned} |F_N(X, Z) - P| &\geq |P - p| - |F_N(X, Z) - p| \\ &\geq \varepsilon - \frac{cK(Z)\sqrt{\log\log Q_N(X, Z)}}{\sqrt{Q_N(X, Z)}}. \end{aligned}$$

Therefore

$$\begin{aligned} &B_N(X, Z, P) \\ &\geq \frac{\left(\varepsilon - \frac{cK(Z)\sqrt{\log\log Q_N(X,Z)}}{\sqrt{Q_N(X,Z)}}\right)\sqrt{Q_N(X, Z)}}{K(Z)\sqrt{\log\log Q_N(X, Z)}} \\ &= \frac{\sqrt{Q_N(X, Z)}}{K(Z)\sqrt{\log\log Q_N(X, Z)}}\varepsilon - c \geq C, \end{aligned}$$

which proves the claim. □

Theorem 5.3. *$A_{L,T}$ passes the generalized Benford test.*

Proof. Let S be an irreducible pattern with probability p. We must show that

$$\lim_{\substack{N\to\infty \\ N\in S}} A_{L,T}(N) = p.$$

Let Z be a Turing machine such that $U(Z, N)$ accepts in time $T(N)$ if and only if $N \in S$.

By considering the case when $X = Z$, Lemma 5.1 implies that there exists a constant C such that for all N sufficiently large, there exists a $P \in J_N$ such that

$$\max_{Y\in TM(N)} \min_{X\in TM(N)} B_N(X, Y, P) < C.$$

Similarly, using this value of C, and considering the case where $Y = Z$, Lemma 5.2 implies that for all $\varepsilon > 0$, for all N sufficiently large, for all $P \in J_N$ if $N \in S$, and

$$\max_{Y\in TM(N)} \min_{X\in TM(N)} B_N(X, Y, P) < C,$$

then $|P - p| \leq \varepsilon$.

Combining these, we get that for all $\varepsilon > 0$, for all N sufficiently large, if $N \in S$ and if P is in

$$\arg\min_{P\in J_N} \max_{Y\in TM(N)} \min_{X\in TM(N)} B_N(X, Y, P),$$

then $|P - p| \leq \varepsilon$.

Thus, by Lemma 4.1, we get that for all $\varepsilon > 0$, for all N sufficiently large, if $N \in S$, then $|A_{L,T}(N) - p| \leq \varepsilon$, so

$$\lim_{\substack{N\to\infty \\ N\in S}} A_{L,T}(N) = p.$$

□

6 Final Remarks

Definition 6.1. *Given a sentence ψ, consider the infinite sequence of integers $\{s_n^\psi\}$ given by $\phi_{s_0^\psi} = \psi$ and $\phi_{s_{n+1}^\psi} = \neg\neg\phi_{s_n^\psi}$. If a machine M satisfies*

$$\lim_{n\to\infty} M(s_n^\psi) = p,$$

we say that M converges to p on ψ.

Corollary 6.2. *If ψ is provable, then $A_{L,T}$ converges to 1 on ψ. If ψ is disprovable, then $A_{L,T}$ converges to 0 on ψ.*

Proof. If ψ is provable, then $\{s_n^\psi\}$ is an irreducible pattern with probably 1. If ψ is disprovable, then $\{s_n^\psi\}$ is an irreducible pattern with probably 0. □

If ψ is neither provable nor disprovable, then it is not clear whether or not $A_{L,T}$ even converges on ψ.

Question 6.3. *Does there exist a machine M such that M passes the generalized Benford test, and for each sentence ψ, there exists a $P(\psi)$ such that M converges to $P(\psi)$ on ψ?*

Definition 6.4. *A function P from logical sentences to $[0, 1]$ is called coherent if it satisfies the following three properties:*

1. *$P(\phi) = 1$ for all provable ϕ,*
2. *$P(\phi) = 0$ for all disprovable ϕ, and*
3. *$P(\phi) = P(\phi \wedge \psi) + P(\phi \wedge \neg\psi)$ for all ϕ and ψ.*

Coherent functions correspond to probability distributions on the space of complete extensions of a given theory.

Question 6.5. *Does there exist a machine M and a coherent function P such that M passes the generalized Benford test, and for each sentence ψ, M converges to $P(\psi)$ on ψ?*

References

[1] Paul Christiano. *Non-Omniscience, Probabilistic Inference, and Metamathematics*. Tech. rep. 2014–3. Berkeley, CA: Machine Intelligence Research Institute, 2014. URL: http://intelligence.org/files/Non-Omniscience.pdf.

[2] Abram Demski. "Logical Prior Probability". In: *Artificial General Intelligence. 5th International Conference, AGI 2012, Oxford, UK, December 8–11, 2012. Proceedings*. Lecture Notes in Artificial Intelligence 7716. New York: Springer, 2012, pp. 50–59. DOI: 10.1007/978-3-642-35506-6_6.

[3] Haim Gaifman. "Reasoning with Limited Resources and Assigning Probabilities to Arithmetical Statements". In: *Synthese* 140.1–2 (2004), pp. 97–119. DOI: 10.1023/B:SYNT.0000029944.99888.a7.

[4] Nate Soares and Benja Fallenstein. "Aligning Superintelligence with Human Interests. A Technical Research Agenda". In: *The Technological Singularity: Managing the Journey*. Ed. by Jim Miller et al. Vol. 2. Springer, forthcoming.

[5] Haim Gaifman. "Concerning Measures in First Order Calculi". In: *Israel Journal of Mathematics* 2.1 (1964), pp. 1–18. DOI: 10.1007/BF02759729.

[6] Marcus Hutter et al. "Probabilities on Sentences in an Expressive Logic". In: *Journal of Applied Logic* 11.4 (2013), pp. 386–420. ISSN: 1570-8683. DOI: http://dx.doi.org/10.1016/j.jal.2013.03.003. URL: http://www.sciencedirect.com/science/article/pii/S157086831300013X.

[7] Dana Scott and Peter Krauss. "Assigning probabilities to logical formulas". In: *Studies in Logic and the Foundations of Mathematics* 43 (1966), pp. 219–264.

[8] Alfred Horn and Alfred Tarski. "Measures in Boolean algebras". In: *Transactions of the American Mathematical Society* 64.3 (1948), pp. 467–467. ISSN: 0002-9947. DOI: 10.1090/S0002-9947-1948-0028922-8. URL: http://www.ams.org/journals/tran/1948-064-03/S0002-9947-1948-0028922-8/S0002-9947-1948-0028922-8.pdf.

[9] J. L. Kelley. "Measures on Boolean algebras". In: *Pacific Journal of Mathematics* 9.4 (1959), pp. 1165–1177. ISSN: 0002-9939. DOI: 10.1090/S0002-9939-1991-1050019-X.

[10] Dorothy Maharam. "An algebraic characterization of measure algebras". In: *Annals of Mathematics* 48.1 (1947), pp. 154–167. ISSN: 01960644. DOI: 10.1016/j.annemergmed.2010.11.022.

[11] Nate Soares and Benja Fallenstein. *Questions of Reasoning Under Logical Uncertainty*. Tech. rep. 2015–1. Berkeley, CA: Machine Intelligence Research Institute, 2015. URL: https://intelligence.org/files/QuestionsLogicalUncertainty.pdf.

[12] Lise Getoor. *Introduction to statistical relational learning*. MIT press, 2007.

[13] L. Pietronero et al. "Explaining the uneven distribution of numbers in nature: the laws of Benford and Zipf". In: *Physica A: Statistical Mechanics and its Applications* 293.1-2 (2001), pp. 297–304. ISSN: 03784371. DOI: 10.1016/S0378-4371(00)00633-6. arXiv: 9808305 [cond-mat].

[14] Ker-I Ko. "On the notion of infinite pseudorandom sequences". In: *Theoretical Computer Science* 48 (1986), pp. 9–33. ISSN: 03043975. DOI: 10.1016/0304-3975(86)90081-2.

[15] Per Martin-Löf. "The definition of random sequences". In: *Information and Control* 9.6 (1966), pp. 602–619. ISSN: 00199958. DOI: 10.1016/S0019-9958(66)80018-9.

[16] Rodney G. Downey and Denis R. Hirschfeldt. *Algorithmic randomness and complexity*. Springer Science & Business Media, 2010. ISBN: 9780387955674. DOI: 10.4249/scholarpedia.2574.

[17] F. C. Hennie and R. E. Stearns. "Two-tape simulation of multitape Turing machines". In: *Journal of the ACM* 13.4 (1966), pp. 533–546. ISSN: 00045411. DOI: 10.1145/321356.321362.