

---

# (Non-)Equivalence of Universal Priors

---

Ian Wood<sup>1</sup> and Peter Sunehag<sup>1</sup> and Marcus Hutter<sup>1,2</sup>

Research School of Computer Science

<sup>1</sup>Australian National University and <sup>2</sup>ETH Zürich

{ian.wood,peter.sunehag,marcus.hutter}@anu.edu.au

15 November 2011

## Abstract

Ray Solomonoff invented the notion of universal induction featuring an aptly termed “universal” prior probability function over all possible computable environments [Sol64]. The essential property of this prior was its ability to dominate all other such priors. Later, Levin introduced another construction — a mixture of all possible priors or “universal mixture” [ZL70]. These priors are well known to be equivalent up to multiplicative constants. Here, we seek to clarify further the relationships between these three characterisations of a universal prior (Solomonoff’s, universal mixtures, and universally dominant priors). We see that the the constructions of Solomonoff and Levin define an identical class of priors, while the class of universally dominant priors is strictly larger. We provide some characterisation of the discrepancy.

## Contents

<b>1</b>	<b>Introduction</b>	<b>2</b>
<b>2</b>	<b>Definitions</b>	<b>2</b>
<b>3</b>	<b>Equivalence between Solomonoff priors and universal mixtures</b>	<b>5</b>
<b>4</b>	<b>Not all universally dominant enumerable semimeasures are universal mixtures</b>	<b>8</b>
<b>5</b>	<b>Conclusions</b>	<b>9</b>
	<b>References</b>	<b>10</b>

## Keywords

algorithmic information theory; universal induction; universal prior.

# 1 Introduction

In the study of universal induction, we consider an abstraction of the world in the form of a binary string. Any sequence from a finite set of possibilities can be expressed in this way, and that is precisely what contemporary computers are capable of analysing. An “environment” provides a measure of probability to (possibly infinite) binary strings. Typically, the class  $\mathcal{M}$  of enumerable semimeasures is considered. Given the equivalence between  $\mathcal{M}$  and the set of monotone Turing machines (Lemma 6), this choice reflects the expectation that the environment can be computed by (or at least approximated by) a Turing machine.

Universal induction is an ideal Bayesian induction mechanism assigning probabilities to possible continuations of a binary string. In order to do this, a prior distribution, termed a universal prior, is defined on binary strings. This prior has the property that the Bayesian mechanism converges to the true (generating) environment for *any* environment  $\mu$  in  $\mathcal{M}$ , given sufficient evidence.

There are three popular ways of defining a universal prior in the literature: Solomonoff’s prior [Sol64, ZL70, Hut05], as a universal mixture [ZL70, Hut05, Hut07], or a universally dominant semimeasure [Hut05, Hut07]. Briefly, a universally dominant semimeasure is one that dominates every other semimeasure in  $\mathcal{M}$  (Definition 9), a universal mixture is a mixture of all semimeasures in  $\mathcal{M}$  with non-zero coefficients (Definition 8), and a Solomonoff prior assigns the probability that a (chosen) monotone universal Turing machine outputs a string given random input (Definition 7). These and other relevant concepts are defined in more detail in Section 2.

Solomonoff’s and the universal mixture constructions have been known for many years and they are often used interchangeably in textbooks and lecture notes. Their equivalence has been shown in the sense that they dominate each other [ZL70, Hut05, LV08]. We extend this result in Section 3, showing that they in fact define exactly the same class of priors.

Further, it is trivial to see that both constructions produce universally dominant semimeasures. The converse is, however, not true. Universally dominant semimeasures are a larger class. We provide a simple example to demonstrate this in Section 4.

These results are relatively undemanding technically, however given their fundamental nature, that they have not to our knowledge been published to date, and the relevance to Ray Solomonoff’s famous work on universal induction, we present them here.

The following diagram summarises these inclusion relations:

## 2 Definitions

We represent the set of finite/infinite binary strings as  $\mathbb{B}^*$  and  $\mathbb{B}^\infty$  respectively.  $\epsilon$  denotes the empty string,  $xb$  the concatenation of strings  $x$  and  $b$ ,  $\ell(x)$  the length

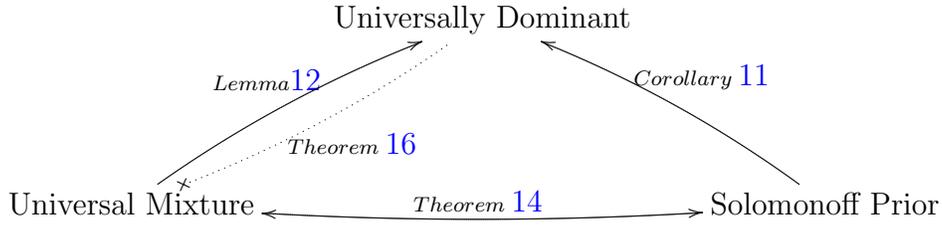


Figure 1:

of a string  $x$ . A cylinder set, the set of all infinite binary strings which start with some  $x \in \mathbb{B}^*$  is denoted  $\Gamma_x$ .

A string  $x$  is said to be a prefix of a string  $y$  if  $y = xz$  for some string  $z$ . We write  $x \sqsubseteq y$  or  $x \sqsubset y$  if  $x$  is a proper substring of  $y$  (ie:  $z \neq \epsilon$ ). We denote the maximal prefix-free subset of a set of finite strings  $\mathcal{P}$  by  $\lfloor \mathcal{P} \rfloor$ . It can be obtained by successively removing elements that have a prefix in  $\mathcal{P}$ . The uniform measure of a set of strings is denoted  $|\mathcal{P}| := \sum_{p \in \lfloor \mathcal{P} \rfloor} 2^{-\ell(p)}$ . This is the area of continuations of elements of  $\mathcal{P}$  considered as binary decimal numbers.

There have been several definitions of monotone Turing machines in the literature [LV08], however we choose that which is now widely accepted [Sol64, ZL70, Hut05, LV08] and has the useful and intuitive property Lemma 6.

**Definition 1.** A monotone Turing machine is a computer with binary (one-way) input and output tapes, a bidirectional binary work tape (with read/write heads as appropriate) and a finite state machine to determine its actions given input and work tape values. The input tape is read-only, the output tape is write-only.

The definitions of a universal Turing machine in the literature are somewhat varied or unclear. Monotone universal Turing machines are relevant here for defining the Solomonoff prior. In the algorithmic information theory literature, most authors are concerned with the explicit construction of a single reference universal machine [Hut05, LV08, Sol64, Tur36, ZL70]. A more general definition is left to a relatively vague statement along the lines of “a Turing machine that can emulate any other Turing machine”. The definition below reflects the typical construction used and is often referred to as *universal by adjunction* [DH10, FSW06].

**Definition 2** (Monotone Universal Turing Machine). A monotone universal Turing machine is a monotone Turing machine  $U$  for which there exist:

1. an enumeration  $\{T_i : i \in \mathbb{N}\}$  of all monotone Turing machines
2. a computable uniquely decodable self-delimiting code  $I : \mathbb{N} \rightarrow \mathbb{B}^*$

such that the programs for  $U$  that produce output coincide with the set  $\{I(i)p : i \in \mathbb{N}, p \in \mathbb{B}^*\}$  of concatenations of  $I(i)$  and  $p$ , and

$$U(I(i)p) = T_i(p) \quad \forall i \in \mathbb{N}, p \in \mathbb{B}^*$$

A key concept in algorithmic information theory is the assignment of probability to a string  $x$  as the probability that some monotone Turing machine produces output beginning with  $x$  given unbiased coin flip input. This approach was used by Solomonoff to construct a universal prior [Sol64]. To better understand the properties of such a function, we will need the concepts of enumerability and semimeasures:

**Definition 3.** A function or number  $\phi$  is said to be *enumerable* or *lower semi-computable* (these terms are synonymous) if it can be approximated from below (pointwise) by a monotone increasing set  $\{\phi_i : i \in \mathbb{N}\}$  of finitely computable functions/numbers, all calculable by a single Turing machine. We write  $\phi_i \nearrow \phi$ . Finitely computable functions/numbers can be computed in finite time by a Turing machine.

**Definition 4.** A *semimeasure* is a “defective” probability measure on the  $\sigma$ -algebra generated by cylinder sets in  $\mathbb{B}^\infty$ . We write  $\mu(x)$  for  $x \in \mathbb{B}^*$  as shorthand for  $\mu(\Gamma_x)$ . A probability measure must satisfy  $\mu(\epsilon) = 1$ ,  $\mu(x) = \sum_{b \in \mathbb{B}} \mu(xb)$ . A semimeasure allows a probability “gap”:  $\mu(\epsilon) \leq 1$  and  $\mu(x) \geq \sum_{b \in \mathbb{B}} \mu(xb)$ .  $\mathcal{M}$  denotes the set of all enumerable semimeasures.

The following definition explicates the relationship between monotone Turing machines and enumerable semimeasures.

**Definition 5** (Solomonoff semimeasure). For each monotone Turing machine  $T$  we associate a semimeasure

$$\lambda_T(x) := \sum_{[p:T(p)=x*]} 2^{-\ell(p)} = |T^{-1}(x*)|$$

where  $[P]$  indicates the maximal prefix-free subset of a set of finite strings  $\mathcal{P}$ ,  $T(p) = x*$  indicates that  $x$  is a prefix of (or equal to)  $T(p)$  and  $\ell(p)$  is the length of  $p$ . If there are no such programs, we set  $\lambda_T(x) := 0$ . [See [LV08] definition 4.5.4]

Note that this is the probability that  $T$  outputs a string starting with  $x$  given unbiased coin flip input. To see this, consider the uniform measure given by  $\lambda(\Gamma_p) := 2^{-\ell(p)}$ . This is the probability of obtaining  $p$  from unbiased coin flips.  $\lambda_T(x)$  is the uniform measure of the set of programs for  $T$  that produce output starting with  $x$ , ie: the probability of obtaining one of those programs from unbiased coin flips. Note also that, since  $T$  is monotone, this set consists of a union of disjoint cylinder sets  $\{\Gamma_p : p \in [q : T(q) = x*]\}$ . By dovetailing a search for such programs and an lower approximation of the uniform measure  $\lambda$ , we can see that  $\lambda_T$  is enumerable. See Definition 4.5.4 (p.299) and Lemma 4.5.5 (p.300) in [LV08].

An important lemma in this discussion establishes the equivalence between the set of all monotone Turing machines and the set  $\mathcal{M}$  of all enumerable semimeasures. It is equivalent to Theorem 4.5.2 in [LV08] (page 301) with a small correction:  $\lambda_T(\epsilon) = 1$  for any  $T$  by construction, but  $\mu(\epsilon)$  may not be 1, so this case must be excluded.

**Lemma 6.** *A semimeasure  $\mu$  is lower semicomputable if and only if there is a monotone Turing machine  $T$  such that  $\mu = \lambda_T$  except on  $\Gamma_\epsilon \equiv \mathbb{B}^\infty$  and  $\mu(\epsilon)$  is lower semicomputable.*

We are now equipped to formally define the 3 formulations for a universal prior:

**Definition 7** (Solomonoff prior). The Solomonoff prior for a given universal monotone Turing machine  $U$  is

$$M := \lambda_U$$

The class of all Solomonoff priors we denote  $\mathcal{U}_M$ .

**Definition 8** (Universal mixture). A universal mixture is a mixture  $\xi$  with non-zero positive weights over an enumeration  $\{\nu_i : i \in \mathbb{N}, \nu_i \in \mathcal{M}\}$  of all enumerable semimeasures  $\mathcal{M}$ :

$$\xi = \sum_{i \in \mathbb{N}} w_i \nu_i \quad : \quad \mathbb{R} \ni w_i > 0, \quad \sum_{i \in \mathbb{N}} w_i \leq 1$$

We require the weights  $w_i$  to be a lower semicomputable function. The mixture  $\xi$  is then itself an enumerable semimeasure, i.e.  $\xi \in \mathcal{M}$ . The class of all universal mixtures we denote  $\mathcal{U}_\xi$ .

**Definition 9** (Universally dominant semimeasure). A universally dominant semimeasure is an enumerable semimeasure  $\delta$  for which there exists a real number  $c_\mu > 0$  for each enumerable semimeasure  $\mu$  satisfying:

$$\delta(x) \geq c_\mu \mu(x) \quad \forall x \in \mathbb{B}^*$$

The class of all universally dominant semimeasures we denote  $\mathcal{U}_\delta$ .

Dominance implies absolute continuity: Every enumerable semimeasure is absolutely continuous with respect to a universally dominant enumerable semimeasure. The converse (absolute continuity implies dominance) is however not true.

### 3 Equivalence between Solomonoff priors and universal mixtures

We show here that every Solomonoff prior  $M \in \mathcal{U}_M$  can be expressed as a universal mixture (i.e.:  $M \in \mathcal{U}_\xi$ ) and vice versa. In other words the class of Solomonoff priors and the class of universal mixtures are identical:  $\mathcal{U}_M = \mathcal{U}_\xi$ .

Previously, it was known [ZL70, Hut05, LV08] that a Solomonoff prior  $M$  and a universal mixture  $\xi$  are equivalent up to multiplicative constants

$$\begin{aligned} M(x) &\leq c_1 \xi(x) & \forall x \in \mathbb{B}^* \\ \xi(x) &\leq c_2 M(x) & \forall x \in \mathbb{B}^* \end{aligned}$$

The result we present is stronger, stating that the two classes are exactly identical. Again we exclude the case  $x = \epsilon$  as  $M(\epsilon)$  is always one for a Solomonoff prior, but  $\xi(\epsilon)$  is never one for a universal mixture  $\xi$  (as there are  $\mu \in \mathcal{M}$  with  $\mu(\epsilon) < 1$ ).

**Lemma 10.** *For any monotone universal Turing machine  $U$  the associated Solomonoff prior  $M$  can be expressed as a universal mixture. i.e. there exists an enumeration  $\{\nu_i\}_{i=1}^{\infty}$  of the set of enumerable semimeasures  $\mathcal{M}$  and computable function  $w_{\square} : \mathbb{N} \rightarrow \mathbb{R}$  such that*

$$M(x) = \sum_{i \in \mathbb{N}} w_i \nu_i(x) \quad \forall x \in \mathbb{B}^* \setminus \epsilon$$

with  $\sum_{i \in \mathbb{N}} w_i \leq 1$  and  $w_i > 0 \forall i \in \mathbb{N}$ . In other words the class of Solomonoff priors is a subset of the class of universal mixtures:  $\mathcal{U}_M \subseteq \mathcal{U}_{\xi}$ .

*Proof.* We note that all programs that produce output from  $U$  are uniquely of the form  $q = I(i)p$ . This allows us to split the sum in (1) below.

$$\begin{aligned} M(x) &= \sum_{[q:U(q)=x^*]} 2^{-\ell(q)} \\ &= \sum_{i \in \mathbb{N}} \sum_{[p:U(I(i)p)=x^*]} 2^{-\ell(I(i)p)} \\ &= \sum_{i \in \mathbb{N}} 2^{-l(I(i))} \sum_{[p:T_i(p)=x^*]} 2^{-\ell(p)} \\ &= \sum_{i \in \mathbb{N}} 2^{-l(I(i))} \lambda_{T_i}(x) \end{aligned} \tag{1}$$

Clearly  $2^{-l(I(i))} > 0$  and is a computable function of  $i$ . Since  $I$  is a self-delimiting code it must be prefix free, and so satisfy Kraft's inequality:

$$\sum_{i \in \mathbb{N}} 2^{-l(I(i))} \leq 1$$

Lemma 6 tells us that the  $\lambda_{T_i}$  cover every enumerable semimeasure if  $\epsilon$  is excluded from their domain, which shows that  $\sum_{i \in \mathbb{N}} 2^{-l(I(i))} \lambda_{T_i}(x)$  is a universal mixture. This completes the proof.  $\square$

**Corollary 11.** *[ZL70] The Solomonoff prior  $M$  for a universal monotone Turing machine  $U$  is universally dominant. Thus, the class of Solomonoff priors is a subset of the class of universally dominant lower semicomputable semimeasures:  $\mathcal{U}_M \subseteq \mathcal{U}_{\delta}$ .*

*Proof.* From Lemma 10 we have for each  $\nu \in \mathcal{M}$  there exists  $j \in \mathbb{N}$  with  $\nu = \lambda_{T_j}$  and for all  $x \in \mathbb{B}^*$ :

$$\begin{aligned} M(x) &= \sum_{i \in \mathbb{N}} 2^{-l(I(i))} \lambda_{T_i}(x) \\ &\geq 2^{-l(I(j))} \nu(x) \end{aligned}$$

as required.  $\square$

**Lemma 12.** *Every universal mixture  $\xi$  is universally dominant. Thus, the class of universal mixtures is a subset of the class of universally dominant lower semicomputable semimeasures:  $\mathcal{U}_\xi \subseteq \mathcal{U}_\delta$ .*

*Proof.* This follows from a similar argument to that in Corollary 11. □

**Lemma 13.** *For every universal mixture  $\xi$  there exists a universal monotone Turing machine and associated Solomonoff prior  $M$  such that*

$$\xi(x) = M(x) \quad \forall x \in \mathbb{B}^* \setminus \epsilon$$

*In other words the class of universal mixtures is a subset of the class of Solomonoff priors:  $\mathcal{U}_\xi \subseteq \mathcal{U}_M$ .*

*Proof.* First note that by Lemma 6 we can find (by dovetailing possible repetitions of some indicies) parallel enumerations  $\{\nu_i\}_{i \in \mathbb{N}}$  of  $\mathcal{M}$  and  $\{T_i = \lambda_{\nu_i}\}_{i \in \mathbb{N}}$  of all monotone Turing machines, and computable weight function  $w_{\langle \rangle}$  with

$$\xi = \sum_{i \in \mathbb{N}} w_i \nu_i \quad , \quad \sum_{i \in \mathbb{N}} w_i \leq 1$$

Take a computable index and lower approximation  $\phi(i, t) \nearrow w_i$ :

$$w_i = \sum_t |\phi(i, t+1) - \phi(i, t)| \tag{2}$$

$$= \sum_j 2^{-k_{ij}} \tag{3}$$

$$i, j \mapsto k_{ij} \text{ computable} \tag{4}$$

The K-C theorem [Lev71, Sch73, Cha75, DH10] says that for any computable sequence of pairs  $\{k_{ij} \in \mathbb{N}, \tau_{ij} \in \mathbb{B}^*\}_{i,j \in \mathbb{N}}$  with  $\sum 2^{-k_{ij}} \leq 1$ , there exists a prefix Turing machine  $P$  and strings  $\{\sigma_{ij} \in \mathbb{B}^*\}$  such that

$$\ell(\sigma_{ij}) = k_{ij} \quad , \quad P(\sigma_{ij}) = \tau_{ij} \tag{5}$$

Choosing distinct  $\tau_{ij}$  and the existence of prefix machine  $P$  ensures that  $\{\sigma_{ij}\}$  is prefix free. We now define a monotone Turing machine  $U$ . For strings of the form  $\sigma_{ij}p$  for some  $i, j$ :

$$U(\sigma_{ij}p) := T_i(p) \tag{6}$$

For strings not of this form,  $U$  produces no output.  $U$  inherits monotonicity from the  $T_i$ , and since  $\{T_i\}_{i \in \mathbb{N}}$  enumerates all monotone Turing machines,  $U$  is universal.

The Solomonoff prior associated with  $U$  is then:

$$\lambda_U(x) = |U^{-1}(x^*)| \tag{7}$$

$$= \sum_{i,j} 2^{-\ell(\sigma_{ij})} |T_i^{-1}(x^*)| \tag{8}$$

$$= \sum_i \left( \sum_j 2^{-k_{ij}} \right) \lambda_{T_i}(x) \tag{9}$$

$$= \sum_i w_i \nu_i(x) \tag{10}$$

$$= \xi(x) \tag{11}$$

□

The main theorem for this section is now trivial:

**Theorem 14.** *The classes  $\mathcal{U}_M$  of Solomonoff priors and  $\mathcal{U}_\xi$  of universal mixtures are exactly equivalent. In other words, the two constructions define exactly the same set of priors:  $\mathcal{U}_M = \mathcal{U}_\xi$ .*

*Proof.* Follows directly from Lemma 10 and Lemma 13. □

## 4 Not all universally dominant enumerable semimeasures are universal mixtures

In this section, we see that a universal mixture must have a “gap” in the semimeasure inequality greater than  $c 2^{-K(\ell(x))}$  for some constant  $c > 0$  independent of  $x$ , and that there are universally dominant enumerable semimeasures that fail this requirement. This shows that not all universally dominant enumerable semimeasures are universal mixtures.

**Lemma 15.** *For every Solomonoff prior  $M$  and associated universal monotone Turing machine  $U$ , there exists a real constant  $c > 0$  such that*

$$\frac{M(x) - M(x0) - M(x1)}{M(x)} \geq c 2^{-K(\ell(x))} \quad \forall x \in \mathbb{B}^*$$

where the Kolmogorov complexity  $K(n)$  of an integer  $n$  is the length of the shortest prefix code for  $n$ .

*Proof.* First, note that  $M(x) - M(x0) - M(x1)$  measures the set of programs  $U^{-1}(x)$  for which  $U$  outputs  $x$  and no more. Consider the set

$$\mathcal{P} := \{ql'p \mid p \in \mathbb{B}^*, U(p) \sqsupseteq x\}$$

where  $l'$  is a shortest prefix code for  $\ell(x)$  and  $q$  is a program such that  $U(ql'p)$  executes  $U(p)$  until  $\ell(x)$  bits are output, then stops.

Now, for each  $r = ql'p \in \mathcal{P}$  we have  $U(r) = x$  since  $U(p) \sqsupseteq x$  and  $q$  executes  $U(p)$  until  $\ell(x)$  bits are output. Thus  $\mathcal{P} \subseteq U^{-1}(x)$  and

$$|\mathcal{P}| \leq |U^{-1}(x)| \quad (12)$$

Also  $\mathcal{P} = ql'U^{-1}(x^*) := \{s = ql'p \mid p \in U^{-1}(x^*)\}$ , and so

$$|\mathcal{P}| = 2^{-\ell(ql')} |U^{-1}(x^*)| \quad (13)$$

combining (12) and (13) and noting that  $M(x) - M(x0) - M(x1) = |U^{-1}(x)|$  and  $M(x) = |U^{-1}(x^*)|$  we obtain

$$\begin{aligned} M(x) - M(x0) - M(x1) &= |U^{-1}(x)| \\ &\geq |\mathcal{P}| \\ &= 2^{-\ell(ql')} |U^{-1}(x^*)| \\ &= 2^{-\ell(q)} 2^{-K(\ell(x))} M(x) \end{aligned}$$

Setting  $c := 2^{-\ell(q)}$  this proves the result.  $\square$

**Theorem 16.** *Not all universally dominant enumerable semimeasures are universal mixtures:  $\mathcal{U}_\xi \subset \mathcal{U}_\delta$*

*Proof.* Take some universally dominant semimeasure  $\delta$ , then define  $\delta'(\epsilon) := 1$ ,  $\delta'(0) = \delta'(1) := \frac{1}{2}$ ,  $\delta'(bx) := \frac{1}{2}\delta(bx)$  for  $b \in \mathbb{B}$ ,  $x \in \mathbb{B}^* \setminus \epsilon$ .  $\delta'$  is clearly a universally dominant enumerable semimeasure with  $\delta'(0) + \delta'(1) = \delta'(\epsilon)$ , and by Lemma 15 it is not a universal mixture.  $\square$

## 5 Conclusions

One of Solomonoff's more famous contributions is the invention of a theoretically ideal universal induction mechanism. The universal prior used in this mechanism can be defined/constructed in several ways. We clarify the relationships between three different definitions of universal priors, namely universal mixtures, Solomonoff priors and universally dominant semimeasures. We show that the class of universal mixtures and the class of Solomonoff priors are exactly the same while the class of universally dominant lower semicomputable semimeasures is a strictly larger set.

We have identified some aspects of the discrepancy between Solomonoff priors/universal mixtures and universally dominant lower semicomputable semimeasures, however a clearer understanding and characterisation would be of interest.

Since universal dominance is all that is needed to prove convergence for universal induction [Hut05, Sol78] it is interesting to ask whether the extra properties of the smaller class of Solomonoff priors have any positive consequences for universal induction.

## Acknowledgements.

We would like to acknowledge the contribution of an anonymous reviewer to a more elegant presentation of the proof of Lemma 13. This work was supported by ARC grant DP0988049.

## References

- [Cha75] G. J. Chaitin. A theory of program size formally identical to information theory. *Journal of the ACM*, 22(3):329–340, 1975.
- [DH10] R. Downey and D. R. Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, Berlin, 2010.
- [FSW06] S. Figueira, F. Stephan, and G. Wu. Randomness and universal machines. *Journal of Complexity*, 22(6):738–751, 2006.
- [Hut05] M. Hutter. *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer, Berlin, 2005.
- [Hut07] M. Hutter. On universal prediction and Bayesian confirmation. *Theoretical Computer Science*, 384(1):33–48, 2007.
- [Lev71] Leonid A Levin. *Some Theorems on the Algorithmic Approach to Probability Theory and Information Theory*. PhD thesis, Moscow University, Moscow, 1971.
- [LV08] M. Li and P. M. B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer, Berlin, 3rd edition, 2008.
- [Sch73] C. P. Schnorr. Process complexity and effective random tests. *Journal of Computer and System Sciences*, 7(4):376–388, 1973.
- [Sol64] R. J. Solomonoff. A formal theory of inductive inference: Parts 1 and 2. *Information and Control*, 7:1–22 and 224–254, 1964.
- [Sol78] R. J. Solomonoff. Complexity-based induction systems: Comparisons and convergence theorems. *IEEE Transactions on Information Theory*, IT-24:422–432, 1978.
- [Tur36] A. M. Turing. On computable numbers, with an application to the Entscheidungsproblem. *Proc. London Mathematical Society*, 2(42):230–265, 1936.
- [ZL70] A. K. Zvonkin and L. A. Levin. The complexity of finite objects and the development of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.