Mutual information and Gödel incompleteness

Diplomarbeit

von

Philipp Geiger

Betreuer:

Dr. Wolfgang Merkle

Oktober 2012

Fakultät für Mathematik und Informatik

Ruprecht-Karls-Universität Heidelberg

Contents

1 Introduction

Peano arithmetic has no *computable* consistent completion - this is the content of Gödel's first incompleteness theorem.[1] It was probably the most surprising result that has been found in the context of Hilbert's program, which calls for a formalization and axiomatization of all of mathematics, together with a "finitary" consistency proof for this axiomatization.[2] It is important to see however, that Gödel's result entails no assertion regarding the *general realizability* of consistent completions of Peano arithmetic. By basic results of mathematical logic, for every consistent axiomatic system there is a consistent completion. Gödel's result only entails the assertion that the consistent completion of Peano arithmetic is impossible by *effectively calculable methods* - if we accept the Church-Turing thesis.

Levin in his paper "Forbidden Information" [Lev10] argues that we can significantly expand this assertion to other than effectively calculable methods. His argumentation can be outlined as follows.

Given any sequence, computing a consistent completion of Peano arithmetic relative to this sequence is equivalent to computing a total extension of the universal partial computable predicate $\mathbf{u}$ relative to this sequence. Due to what we will call the forbidden information theorem, every sequence that computes a total extension of $\mathbf{u}$ has infinite mutual information with the halting probability Ω. If we

[1] Note that the idea behind Gödel's first incompleteness theorem is not only applicable to Peano arithmetic, but to any formal system that has a certain expressive power. Furthermore, note that what Gödel originally proved was a slightly different and weaker theorem (it was extended by Rosser). The version of the first incompleteness theorem we refer to here is the one that is used by Ebbinghaus, Flum and Thomas [EFT94], among others.

[2] For further information on Hilbert's program, see Zach [Zac09].

accept Levin's independence postulate, then no sequence generated by any locatable physical process may have infinite mutual information with Ω. So we can conclude the following extension of Gödel's incompleteness assertion, which we will call the forbidden information thesis: *no sequence that is generated by any locatable physical process is a consistent completion of Peano arithmetic.*

Levin's exposition of the argumentation we just outlined is, however, rather sketchy and difficult to follow in some parts. Moreover, he tends to implicitly use results without explicitly mentioning them or indicating where they were proved.

The *main objective* of the present work is to *completely and critically elaborate Levin's argumentation.* To present a complete argumentation, we gathered up all major underlying mathematical concepts, assertions and proofs (if available) from a variety of sources, discuss them in a detailed manner, and prove some small but necessary assertions ourselves. And to allow a critical assessment, we try to weigh thoroughly the validity of all non-mathematical arguments.

In concrete terms, we will proceed as follows. In the *second chapter* we lay down the foundations we need throughout the work. We clarify notational conventions, introduce the concepts of Kolmogorov complexity, random sequences, and tests. And we will translate the problem of the consistent completion of Peano arithmetic into the computability theoretic problem of finding a total extension of the universal partial computable predicate $\mathbf{u}$.

In the *third chapter*, we give a detailed account of the concept of mutual information, which is the most important instrument Levin uses in "Forbidden Information", and which turns out to be quite interesting by itself. Besides some basic properties, we will present the independence conservation inequalities, which are due to Levin. We then use these inequalities to argue for the independence postulate. This is a non-mathematical assertion (similar to the Church-Turing thesis) which is central to Levin's argumentation. Afterwards we critically discuss the postulate.

While the first two chapters mainly served as preparation, the *fourth and last chapter* contains the actual argumentation in "Forbidden Information". First, we

present the forbidden information theorem and give a more detailed proof than Levin does in his paper. Based on this theorem and the previous results, we will eventually argue for the forbidden information thesis we mentioned above.

Before getting into details, we want so say a few words to justify our interest in the subject-matter of the present work. Generally, there are various purposes of mathematical research, such as solving real-world problems or gaining knowledge which is interesting by itself for mathematical reasons. Another important purpose - which is also one aim of the present work - is to promote knowledge that is interesting by itself since it is *philosophical knowledge*. Keep in mind, that "What can I know?" is the first of three questions that were stated by Kant to describe the central interests of human reason ([Kan29], A805). The forbidden information thesis can be seen as an (attempt of a) partial answer to this question, since it states a definite limit for formal mathematical knowledge.[3] Note, that also Gödel's incompleteness assertion can be philosophically interpreted this way.

[3]It should be mentioned that Kant was mainly looking for *a priori* answers to this question in the "Critique of Pure Reason" [Kan29], whereas the partial answer we discuss in the present work is rather *a posteriori*, since we take into account physical considerations.

2 Preliminaries

We start with clarifying notational conventions and stating basic results we will need throughout the work. The first section mainly contains notational basics that are commonly used. Afterwards, we introduce Kolmogorov complexity and discuss some basic properties. Then we define random sequences and introduce tests as they are defined by Levin. And finally, we will discuss the Church-Turing thesis and translate the problem of finding a consistent completion of Peano arithmetic into the computability theoretic problem of finding a total extension of a universal partial computable predicate. The latter result is the only one from Chapter 2 that will directly appear in the final argumentation in favor of the forbidden information thesis 4.14.

The definitions and theorems of Sections 2.1 through 2.3 are in general taken from the monographs by Li and Vitanyi [LV08], and Downey and Hirschfeldt [DH10], and Levin's 1974 paper [Lev74]. For backgrounds on Section 2.4, we refer to the monographs by Odifreddi [Odi92], Ebbinghaus, Flum and Thomas [EFT94], and Downey and Hirschfeldt [DH10], and the paper "Church's Thesis and Principles for Mechanisms" by Gandy [Gan80].

2.1 Some basics

Besides discussing some basics with respect to sequences, Turing machines and computability, we will explain, what we mean by randomized algorithms and operators.

Throughout the work, we will often consider equalities and inequalites, that hold only up to an additive or multiplicative constant. We will use the relation symbols "$\stackrel{+}{=}$", "$\stackrel{*}{=}$" and "$\stackrel{+}{<}$", "$\stackrel{+}{>}$", "$\stackrel{*}{<}$", "$\stackrel{*}{>}$" in those cases.

2.1.1 Sequences

We denote the set of *binary strings*, which we may also call simply "strings" or "finite (binary) sequences", by $\{0,1\}^*$ and the set of *infinite binary sequences*, which we may also call "infinite sequences" or simply "sequences", by $\{0,1\}^\infty$. We identify infinite sequences, subsets of $\mathbb{N}$, and functions of the form $\mathbb{N} \to \{0,1\}$ in the usual manner. For a finite or infinite sequence a and any $i \in \mathbb{N}$, we denote by $a(i)$ the *symbol at the i-th position of a* and by $a\restriction i$ the *prefix of length i of a*. For example $1110(3) = 0$, $1110\restriction 2 = 11$. For a string $x \in \{0,1\}^*$ we denote its *length* by $\ell(x)$, for example $\ell(1110) = 4$. For two strings $x, y \in \{0,1\}^*$ we denote their *concatenation* by $x^\frown y$ or simply xy. By ϵ, we denote the *empty string*. We generally identify strings and natural numbers using the following bijection

$$\begin{aligned} N\colon \{0,1\}^* &\to \mathbb{N}, \\ x &\mapsto N(x) := 2^{\ell(x)} - 1 + \sum_{0 \le i < \ell(x)} x(i) \cdot 2^i. \end{aligned}$$

Note that by the above definition and identification, for all $n \in \mathbb{N}$

$$\ell(n) = \ell(N^{-1}(n)) \stackrel{+}{=} \log n.$$

By log we mean the *logarithm to base 2*. We fix a *pairing function* $\langle \cdot, \cdot \rangle$ on finite and infinite sequences. For $x, y \in \{0,1\}^*$, $\alpha, \beta \in \{0,1\}^\infty$ let

$$\begin{aligned} \langle x, y \rangle &:= \frac{1}{2}(x+y)(x+y+1) + y, \\ \langle \alpha, \beta \rangle &:= \alpha(0)\beta(0)\alpha(1)\beta(1)\alpha(2)\beta(2)\ldots. \end{aligned}$$

We define the generalizations of the pairing function for arbitrary n-tuples in the usual way (by iterative use of the pairing function) and denote them by $\langle \cdot, \cdot, \ldots, \cdot \rangle_n$.

For the sake of convenience, we may drop the n in the index. For $x \in \{0,1\}^*$ let $\overline{x} := x(0)x(0)x(1)x(1)\dots x(\ell(x)-1)x(\ell(x)-1)01$.

For $x \in \{0,1\}^*$ and $a \in \{0,1\}^* \cup \{0,1\}^\infty$ we mean by $x \sqsubseteq a$ that x is a *prefix* or *initial segment*, respectively, of a. We denote the *cylinder set* defined by x, i.e. $\{\alpha \in \{0,1\}^\infty : x \sqsubseteq \alpha\}$, by $[\![x]\!]$.

2.1.2 Turing machines, computability, prefix-freeness

We define *Turing machines* and *oracle Turing machines* in the usual way. We restrict to the tape alphabet $\{0,1,\square\}$. By $\mathrm{dom}(M)$ we denote the domain of a Turing machine M. As usual, when we talk about the functioning of some oracle machine M, by M^α we mean the machine with the sequence α written on its oracle tape. We also allow strings as oracles but we stipulate that in case the machine attempts to make any queries beyond the length of the string, the computation automatically diverges; so for a fixed input, the finite oracles that yield a terminating computation form a prefix-free set. Keep in mind that a set $S \subset \{0,1\}^*$ is called prefix-free, if for all $x, y \in S$: if $x \sqsubseteq y$, then $x = y$.

By φ_M, we denote the partial function computed by a Turing machine M. For convenience, we mostly write $M(x)$ instead of $\varphi_M(x)$, for any x. If M is an *oracle Turing machine*, Φ_M denotes the partial functional computed by M, i.e. $\Phi_M \colon \{0,1\}^\infty \to \{0,1\}^\infty$. For convenience, we mostly write $M^\alpha(x)$ instead of $\Phi_M(\alpha)(x)$, for any α, x. We call a total function $f \colon \{0,1\}^\infty \to \{0,1\}^\infty$ an *algorithmic operator*, if there is an oracle Turing machine M, such that $f = \Phi_M$.

For a Turing machine M and input x, by $M(x)\downarrow$ we mean that M terminates on input x and by $M(x)\uparrow$ we mean that M does not terminate on input x. For two partial functions φ, ψ on $\mathbb{N}$, and $n \in \mathbb{N}$, by $\varphi(n) \cong \psi(n)$ we mean that either both functions are defined on n and $\varphi(n) = \psi(n)$ or both functions are undefined on n.

We call a (partial) function $f \colon \subseteq \mathbb{N} \to \mathbb{N}$ *(partial) computable*, if $f = \varphi_M$ for some Turing machine M. Given some sequence α, we call f *(partial) computable*

in α or α*(-partial)-computable*, if $f = \Phi_M(\alpha)$, for some oracle Turing machine M. A set $A \subseteq \mathbb{N}$ is called *computable*, if its characteristic function is computable and *computably enumerable*, if it is the domain of a partial computable function. We may say *recursive* instead of computable. We call a function $f\colon \mathbb{N} \to \mathbb{N}$ *lower semicomputable*, if the sets $\{n \in \mathbb{N} : n \leq f(m)\}$ are uniformly computably enumerable in m, and *upper semicomputable*, if $-f$ is lower semicomputable.

We will consider a special kind of Turing machines, namely *prefix-free Turing machines*. We call a Turing machine M prefix-free, if $\mathrm{dom}(\varphi_M)$ is prefix-free. Furthermore, we call an oracle Turing machine M prefix-free, if for any finite oracle $x \in \{0,1\}^*$, $\mathrm{dom}(\Phi_M(\overline{x}))$ is prefix-free and for any infinite oracle $\alpha \in \{0,1\}^\infty$, $\mathrm{dom}(\Phi_M(\alpha))$ is prefix-free. Similar to the well-known universal oracle Turing machines, there is a *universal prefix-free oracle Turing machine* which we denote by $\mathbf{U}$.[1] More precisely, for all prefix-free oracle Turing machines M, there is a string p_M, such that

$$\mathbf{U}^a(p_M{}^\frown x) \cong M^a(x), \quad \text{for all } a \in \{0,1\}^* \cup \{0,1\}^\infty, x \in \{0,1\}^*.$$

Note that there are (infinitely) many universal prefix-free oracle Turing machine, but we picked out one.

2.1.3 Computable reals, functions and measures

By *reals* we generally mean elements of $[0,1]$. We may identify an infinite binary sequence α with the real $0.\alpha$ in $[0,1]$ (obviously this is not an actual bijection, as there are reals with more than one binary representations; but that does not matter for our purposes). We call a real r *computable*, if r may be binary represented by $0.\alpha$, for some computable $\alpha \in \{0,1\}^\infty$. We say a real r is *left-computably enumerable* (or *left-c.e.*), if there is a computable, monotone increasing sequence

[1] For further details on the construction of the universal prefix-free (oracle) Turing machine, which makes use of a correspondence between prefix-free and so-called self-delimiting (oracle) Turing machines, see for example Downey and Hirschfeldt [DH10].

$(q_i)_{i\in\mathbb{N}} \subset \mathbb{Q}$ such that $\lim_{i\to\infty} q_i = r$. (We represent rational numbers by natural numbers using the pairing function.)

Let us turn to real-valued functions. We call a function $f\colon \{0,1\}^* \to [0,1]$ *computable*, if there is a computable, rational-valued function $g(\cdot,\cdot)$, such that $|f(x) - g(x,k)| < \frac{1}{k}$, for all k. We call $f\colon \{0,1\}^* \to [0,1]$ *lower semicomputable* (or *left-c.e.*), if there is a computable, rational-valued function $g(\cdot,\cdot)$, such that $g(x,k) \leq g(x,k+1)$, for all x,k, and $\lim_{k\to\infty} g(x,k) = f(x)$, for all x. (Note that this is more or less a generalization of lower semicomputability for integer functions.)

The important real-valued functions we consider are measures. It should be clear now, what we mean by a computable (or lower semicomputable) measure on $\{0,1\}^*$, which we call *discrete*. But how to treat measures on $\{0,1\}^\infty$, which we call *continuous*, in terms of computability?

First note, that we restrict our attention to measures on the Borel σ-algebra B generated by the set of cylinder sets, i.e. $\{[\![x]\!] : x \in \{0,1\}^*\}$. (Keep in mind that when we speak of "measures on $\{0,1\}^\infty$" we actually mean measures on B.) B is uncountable, so how can functions on B be anyhow "computable"? Let us consider the set R of all finite unions of cylinder sets, which is a ring (in the set-theoretic sense). By basic results of measure theory, any measure P on B is already uniquely determined by its values on R. But the values of P on R are already determined by its values on the cylinder sets.

Therefore, we simply say that a probability measure P on B is *computable*, if the mapping $x \mapsto P([\![x]\!])$ is computable. Although P has an uncountable domain, this way of speaking makes sense, since by the above consideration the algorithm that computes the mapping $x \mapsto P([\![x]\!])$ fully determines P.

2.1.4 Randomized algorithms and randomized operators

For the interpretation of the independence conservation inequalities in Chapter 3, we will need the notions of randomized algorithms (for strings) and randomized

operators (the analogon for sequences). In what follows, we will particularly proof the computability of the output probability distribution for such algorithms and operators.

Randomized algorithms

By a *randomized algorithm* we mean a (total) Turing machine M, that has an auxiliary "randomness tape". The randomness tape has infinite sequences written on it[2] and we consider these sequences distributed according to a computable probability distribution Q on $\{0,1\}^\infty$. Note that we may consider the pair (M, Q) instead of only M as the randomized algorithm. We assume that for any input and any random sequence, M terminates.

We will be particularly interested in the probability that M computes y on input x, which we denote by $P_x(y)$ for the moment. With an argumentation similar to the one for (compositions of) randomized operators (see the following section), it is easy to see that $(x, y) \mapsto P_x(y)$ is computable in the sense of Section 2.1.3.

Randomized operators on infinite sequences

Let us now turn to randomized operators on infinite sequences, which should be seen as an analogon to randomized alogorithms on finite sequences. We define a *randomized operator* as an oracle machine M, that gets the input sequence written on the oracle tape, a random sequence written on an auxiliary randomness tape and that step-by-step writes an infinite sequence on the output tape (by running infinitely long). (Obviously we deviate a bit from the standard definition of the Turing machine by allowing infinite output sequences but this is necessary.) As above, we consider the random sequence as distributed according to a computable probability distribution Q. Again, we may consider the pair (M, Q) instead of only M as the randomized operator.

[2] In many cases, only randomized algorithms that have access to a fixed number of (uniformly distributed) random bits are considered. But having in mind the very general claim of the independence postulate in Chapter 4, we want to consider a more general concept.

In Section 3.3 we will work with a finite *composition of randomized operators*. By such a composition we mean that one operator writes its infinite output on the oracle tape of the next one. (Note that every operator of the composition has its own randomness tape.) We will be particularly interested in the probability that the composition of operators, on input α, computes a sequence with the initial segment z, which we denote by $P_\alpha([\![z]\!])$ for the moment. By the following simple argumentation, $P_\alpha([\![z]\!])$ is α-computable.

We argument by induction. Let n be the number of random operators the composition consists of. We assume that $P'_\alpha([\![w]\!])$, the probability that the composition of the first $n-1$ operators outputs a sequence starting with the prefix w, on input α, is α-computable. Let (M, Q) be the n-th operator.

Let S_z denote the set of pairs $\langle x, y\rangle$, such that at some point of time M hast just written the last bit of z on the output tape, after it has exactly read x on the oracle tape and y on the randomness tape. Let

$$\tilde{S}_l := \{\langle x, y\rangle : \ \langle x, y\rangle \in S_z, \text{ for some } z \text{ with } \ell(z) = l\}.$$

By König's lemma, $\tilde{S}_{\ell(z)}$ is finite for all z and uniformly computable in z. Therefore, S_z is finite and uniformly computable in z. Furthermore, note that for all z and $\langle x, y\rangle, \langle x', y'\rangle \in S_z$, x is incomparable to x' (i.e. one can be no prefix of the other) and y is incomparable to y'.

Hence, for all z we have[3]

$$P_\alpha([\![z]\!]) = P'_\alpha \otimes Q \left(\bigcup_{\langle x,y\rangle \in S_z} [\![x]\!] \times [\![y]\!] \right) = \sum_{\langle x,y\rangle \in S_z} P'_\alpha([\![x]\!]) \cdot Q([\![y]\!]).$$

So $P_\alpha([\![z]\!])$ is α-computable.

[3]We take the product measure $P'_\alpha \otimes Q$ since it is the unique measure on the product space that makes the input sequence and the random sequence stochastically independent (we assume their independence) and that preserves the measures P'_α and Q on "their spaces".

2.2 Kolmogorov Complexity

We turn to Kolmogorov complexity now, which is considered in algorithmic information theory. This is the first topic we want to discuss more detailed as it will play a central role throughout this work. The aim is to quantize information based on algorithmic considerations. It should be mentioned, that Kolmogorov complexity serves as a central tool in the proof of the forbidden information theorem 4.1 and thus also for our argumentation in favor of an extension of Gödel's incompleteness assterion, which culminates in the forbidden information thesis 4.14.

The general idea behind measures of information (also within Shannon's probabilistic information theory[4]) is the following: We have a fixed set of possible messages from which one message is chosen and thereby actual information is produced. Furthermore, we have a description system (or "code") for our set of possible messages, consisting of finite sequences over a fixed alphabet. Now the quantity of information contained in a selected message is defined as the optimal[5] length of a description for the message (or stated differently, the optimal number of choices to determine the message using the symbols of the fixed alphabet).

In our case, i.e. Kolmogorov complexity, we implement this general idea as follows. The fixed set of possible message is the set of binary strings. Our description system consists (again) of binary strings, from which we, by algorithmical means, can construct the original messages. Now the optimal description for a selected messages is simply interpreted as the shortest one.[6]

Note that in the history of the field of algorithmic information theory, several attempts were made regarding the precise definition of an algorithmic measure

[4]For further information, see MacKay [Mac03].

[5]In probabilistic information theory, this optimality is interpreted in the "stochastically global" sense, i.e. if the messages are chosen according to a fixed probability distribution, then the *expected length* of the corresponding codewords should be shortest possible. Note that the *entropy* is simply the expected codeword length for an optimal code for the distribution, e.g. the Shannon-Fano code. For further details on this topic, see MacKay [Mac03].

[6]We can approach Kolmogorov complexity from a more practical direction, too. Most people that use computers should be familiar with "ZIP" programs, i.e. data compression software. Kolmogorov complexity gives us a lower bound for how far we can compress given data.

of information (such as plain Kolmogorov complexity, monotone complexity). In the present work, we will only use prefix Kolmogorov complexity. Note that for the sake of simplicity, we will generaly drop the "prefix" and only speak of "Kolmogorov complexity"!

2.2.1 Definition

As already mentioned, the Kolmogorov complexity of a given string is roughly speaking the length of the shortest codeword from which we can algorithmically reconstruct the string. Now we first define plain Kolmogorov complexity, but only as a means to define (prefix) Kolmogorov complexity afterwards.

Definition 2.1. Let M be a Turing machine and $x, y \in \{0,1\}^*$. We define

$$\mathrm{C}_M(x|y) := \min\{\ell(p) : M^{\overline{y}}(p) = x\},$$

which we call *plain Kolmogorov complexity of x relative to y, with respect to M*, with $\min \emptyset := \infty$.
We write $\mathrm{C}_M(x) = \mathrm{C}_M(x|\epsilon)$ and call it *plain Kolmogorov complexity of x, with respect to M*.

We also want to consider infinite sequences in the conditional part of plain Kolmogorov complexity, so we immediately extend the definition.

Definition 2.2. Let M be an oracle Turing machine and $x \in \{0,1\}^*$, $\alpha \in \{0,1\}^\infty$. We define

$$\mathrm{C}_M(x|\alpha) := \min\{\ell(p) : M^{\alpha}(p) = x\},$$

with $\min \emptyset := \infty$.

As already mentioned, we only work with prefix Kolmogorov complexity in the present work, where we generally drop the "prefix". For its precise definition, we need the universal prefix-free oracle Turing machine $\mathbf{U}$ (defined in Section 2.1.2).

Definition 2.3. Let $x \in \{0,1\}^*$, $a \in \{0,1\}^* \cup \{0,1\}^\infty$. We define the *Kolmogorov complexity of x relative to a* as

$$\mathbf{K}(x|a) := C_{\mathbf{U}}(x|a).$$

We write $\mathbf{K}(x) := \mathbf{K}(x|\epsilon)$.

Remark 2.4. For prefix-free Turing machines M we sometimes also write $\mathbf{K}_M$ instead of C_M.

Remark 2.5. Note the important fact, that for any other universal prefix-free oracle Turing machine U we have

$$\mathbf{K}(x|a) \stackrel{+}{=} \mathbf{K}_U(x|a), \quad \text{for all } x \in 0,1^*, a \in \{0,1\}^* \cup \{0,1\}^\infty$$

(which is more or less due to Theorem 2.9 below). So our definition does depend on the universal prefix-free oracle Turing machine we fixed only up to an additive constant.

We extend the definition of Kolmogorov compexity to tuples of strings.

Definition 2.6. For $x_1, x_2, \ldots, x_m \in \{0,1\}^*$ and $a_1, a_2, \ldots, a_n \in \{0,1\}^* \cup \{0,1\}^\infty$ we define

$$\mathbf{K}(x_1, x_2, \ldots, x_m | a_1, a_2, \ldots, a_n) := \mathbf{K}(\langle x_1, x_2, \ldots, x_m \rangle | \langle a_1, a_2, \ldots, a_n \rangle).$$

We want to fix some optimal description for a given string.

Definition 2.7. For strings $x, y \in \{0,1\}^*$ the operator $*(y)$ is defined by $x^{*(y)}$ being the string p for which the calculation of $\mathbf{U}^{\overline{y}}(p)$ first diverges, among all p with $\mathbf{U}^{\overline{y}}(p) = x$ and $\ell(p) = \mathbf{K}(x|y)$. Furthermore, let $x^* := x^{*(\epsilon)}$.

2.2.2 Basic properties of prefix Kolmogorov complexity

We proceed with presenting some basic statements about Kolmogorov complexity. First, we want to state one fundamental fact about Kolmogorov complexity that

weakens its practicability a bit.

Theorem 2.8. *The function* $(x, y) \mapsto \mathbf{K}(x|y)$ *is not computable, though upper semicomputable i.e., the sets* $\{n \in \mathbb{N} : n \geq \mathbf{K}(x|y)\}$ *are uniformly computably enumerable in* (x, y)*.*

It immediately follows from the universality of $\mathbf{U}$ that $\mathbf{K}$, as defined above, is, up to an additive constant, minimal.

Theorem 2.9. *If* M *is a prefix-free oracle Turing machine, then there is a constant* c_M*, such that for all* $x \in 0, 1^*,\ a \in \{0,1\}^* \cup \{0,1\}^\infty$ *we have*

$$\mathbf{K}(x|a) \leq \mathbf{K}_M(x|a) + c_M.$$

For any string x, we can obviously compute $(x, \mathbf{K}(x))$ from x^*, and vice versa, so the following equality holds.

Theorem 2.10. *For all* $x \in \{0,1\}^*,\ a \in \{0,1\}^* \cup \{0,1\}^\infty$ *we have*

$$\mathbf{K}(x^*|a) \stackrel{+}{=} \mathbf{K}(x, \mathbf{K}(x)|a).$$

Let us now state a "triangle inequalitiy".

Theorem 2.11. *Let* $x, y, z \in \{0,1\}^*$*. Then we have*

$$\mathbf{K}(x|y) \stackrel{+}{<} \mathbf{K}(x, z|y) \stackrel{+}{<} \mathbf{K}(x|z) + \mathbf{K}(z|y).$$

Proof. The first inequality is obvious. Regarding the the second one: From $z^{*(y)\frown}x^{*(z)}$ and oracle $\overline{y}$ we can first compute z and then also x (we can construct a prefix-free oracle machine $M^{\overline{y}}$ that splits up $z^{*(y)\frown}x^{*(z)}$, using $\mathbf{U}^{\overline{y}}$ and the prefix-freeness of its domain). Moreover $\ell(z^{*(y)\frown}x^{*(z)}) = \mathbf{K}(x|z) + \mathbf{K}(z|y)$.

□

We state two inequalities regarding $\mathbf{K}$.

Theorem 2.12. *The following inequalities hold true for all* $x, y \in \{0,1\}^*, a \in \{0,1\}^* \cup \{0,1\}^\infty$*:*

(i) $\mathbf{K}(x,y|a) \overset{+}{<} \mathbf{K}(x|a) + \mathbf{K}(y|a)$.

(ii) If f *is a computable function, then* $\mathbf{K}(f(x)|a) \overset{+}{<} \mathbf{K}(x|a)$.

Now we present two upper bounds for prefix Kolmogorov complexity.

Theorem 2.13. *For all* $x \in \{0,1\}^*$ *the following inequalities hold true:*

(i) $\mathbf{K}(x) \overset{+}{<} 2\,\ell(x)$.

(ii) $\mathbf{K}(x) \overset{+}{<} \ell(x) + \mathbf{K}(\ell(x))$.

The following theorem will be important for the next section. Have in mind the famous Kraft-McMillan theorem, which establishes a tight link between prefix-free codes and mass functions. It states that, given some set of strings $A = \{a_0, a_1, \ldots\}$, the following holds true: "If C is a prefix-free binary code for A, then $\sum_{x\in A} 2^{\ell(C(x))} \leq 1$. Conversely, if $(l_0, a_0), (l_1, a_1), \ldots$ is a sequence of pairs of codeword lengths and strings satisfying *Kraft's inequality*, i.e. $\sum_{i\in\mathbb{N}} 2^{l_i} \leq 1$, then there is prefix-free code C, such that $\ell(C(a_i)) = l_i$, for all i." What follows now is an effective version of the second direction of the Kraft-McMillan theorem.

Theorem 2.14 (KC theorem)**.** *Let* $(d_i, x_i)_{i\in\mathbb{N}} \subset (\mathbb{N} \times \{0,1\}^*)^{\mathbb{N}}$ *be an* A*-computable sequence, for some* $A \subset \mathbb{N}$*, such that*

$$\sum_{i\in\mathbb{N}} 2^{-d_i} \leq 1.$$

Then there is a prefix-free oracle Turing machine M*, and a prefix-free set of strings* $\{y_1, y_2, \ldots\}$ *so that* $\ell(y_i) = d_i$ *and* $M^A(y_i) = x_i$*, for all* $i \in \mathbb{N}$.

2.2.3 The universal discrete semimeasure

There is a different approach towards Kolmogorov complexity, that mainly Levin takes (see e.g. [Lev84], [Lev10]). He first introduces the so-called universal discrete semimeasure as a universal element of the set of all discrete semimeasures (see Definition 2.15 below) and based upon this, he defines Kolmogorov complexity and particularly mutual information, which we will discuss later. Though we do not follow Levin in taking the universal discrete semimeasure as basic term, we will make some use of it throughout this work, so we shortly want to discuss it now.

Definition 2.15. A function $f\colon \{0,1\}^* \to [0,1]$ is called a *discrete semimeasure* if

$$\sum_{x\in\{0,1\}^*} f(x) \leq 1.$$

A lower semicomputabel discrete semimeasure which multiplicatively, up to a constant factor, majorizes every other lower semicomputabel semimeasure is called *universal*. We may call such a semimeasure simply *universal measure*, as well.

Not too surprising, since we deal with a family of (semi-)computable functions, in fact there is a universal measure.[7]

Theorem 2.16 (Existence of the universal measure)**.** *There is a universal lower semicomputable discrete semimeasure.*

Within this section, let m denote the universal measure whose existence is guaranteed by the above theorem.

Remark 2.17 (Output probability Q). There is an interesting function on $\{0,1\}^*$, which we call Q and which can be closely linked to $\mathbf{K}$ and m (see Theorem 2.18

[7] Note that the fundamental idea of a "universal object" that is included in Turing's universal Turing machine has a wide range of implementations in this work (we will define the so-called universal partial computable predicate later). Levin seems to have been strongly inspired by this fundamental idea.

below): the "probability" that our universal prefix-free Turing machine $\mathbf{U}$ outputs some string x. We can define it by

$$Q(x) := Q_{\mathbf{U}}(x) := \sum_{p:\mathbf{U}(p)=x} 2^{\ell(y)} = \lambda([\![\{p \in \{0,1\}^* : \mathbf{U}(p) = x\}]\!]),$$

based on the idea, that we declare the probability of $p \in \mathrm{dom}(\mathbf{U})$ to be $2^{\ell(p)}$; λ denotes the uniform distribution on $\{0,1\}^\infty$, i.e. the measure that is (uniquely) determined by $\lambda([\![x]\!]) = 2^{-\ell(x)}$, for all x.

Now we present the promised link between $\mathbf{K}$, m and Q. Its proof is heavily based on the KC theorem 2.14.

Theorem 2.18. *We have*

$$m(x) \stackrel{*}{=} 2^{-\mathbf{K}(x)} \stackrel{*}{=} Q(x).$$

For practical reasons we will, instead of the above defined m, from now on fix the following function $\mathbf{m}$ as universal measure, extending the original definition 2.15 of a universal measure a litte bit by allowing a conditional argument.

Definition 2.19 (Universal lower semicomputable discrete semimeasure)**.** Let us denote

$$\mathbf{m}(x_1, \ldots, x_m | a_1, \ldots, a_n) := 2^{-\mathbf{K}(x_1,\ldots,x_m|a_1,\ldots,a_n)},$$

for all $x_1, \ldots, x_m \in \{0,1\}^*,\ a_1, \ldots, a_n \in \{0,1\}^* \cup \{0,1\}^\infty$.
We write $\mathbf{m}(x) := \mathbf{m}(x|\epsilon)$, for any x. We call $\mathbf{m}$ *universal lower semicomputable discrete semimeasure*, or *universal measure*.

Remark 2.20. Obviously, we now have an even more general version of a discrete semimeasure, namely one with a conditional argument. It is important to see that for any a, again,

$$\sum_{x \in \{0,1\}^*} \mathbf{m}(x|a) \leq 1,$$

due to the fact that the $\mathbf{K}(x|a)$'s are the codeword lengths of a prefix-free code (namely the domain of $\mathbf{U}^a$) and Kraft's inequality.

Furthermore we remark, without proof, that $\mathbf{m}$ is again multiplicatively, up to a constant factor, majorizing all lower semicomputable functions $f\colon \{0,1\}^*\times\{0,1\}^*\cup\{0,1\}^\infty \to [0,1]$ with $\sum_{x\in\{0,1\}^*} f(x,a) \leq 1$ for any a.

For the more general universal measure $\mathbf{m}$ we just defined, we have a more general version of Theorem 2.18, as well.

Theorem 2.21 (Coding theorem)**.** *We have*

$$\mathbf{m}(x|a) \stackrel{*}{=} 2^{-\mathbf{K}(x|a)},$$

for all $x \in \{0,1\}^*,\ a \in \{0,1\}^* \cup \{0,1\}^\infty$.

2.3 Random sequences and Levin's tests

In this section we want to discuss the concept of a random sequence on the one hand, and tests as Levin defines them on the other hand. We will need a special kind of random sequence in the forbidden information theorem 4.1 and for the justification of the forbidden information thesis 4.14.

In many cases, (randomness) tests are used solely to define random sequences, but not in this work - we define random sequences using Kolmogorov complexity. We introduce the concept of a test mainly as a means to express the independence conservation inequalities in Chapter 3. The tests we will define are due to Levin and differ form the well-known randomness tests which are due to Martin-Löf. However, both kinds of tests are similar and, for the sake of a better understanding, we will use Martin-Löf tests to establish a link between tests and random sequences. We will define tests for both, infinite and finite sequences.

2.3.1 Random sequences

Given a large amount of data, it seems natural to regard this data as random, if we can not recognize any law behind it. Considering a very long but finite binary

sequence, we may rephrase this as follows: The particular sequence is random, if its shortest description is not essentially shorter than the sequence itself (i.e., we basically have to take the sequence itself as its shortest description). Using Kolmogorov complexity, we can express this by $\mathbf{K}(x) \approx \ell(x)$, if we denote our random sequence by x.

We may consider an infinite sequence as random, if all initial segments are random. Based on these considerations, we make the following definition.

Definition 2.22 (Martin-Löf random sequence)**.** A sequence $\alpha \in \{0,1\}^\infty$ is called *Martin-Löf random*, if $\sup_{n\in\mathbb{N}} n - \mathbf{K}(\alpha\restriction n) < \infty$.

2.3.2 Tests for infinite sequences

Martin-Löf gave the following definition for randomness tests. The idea behind this definition is, that a random sequence should satisfiy all effective laws of probability one (which correspond to randomness tests), such as the famous law of large numbers. Note that λ denotes the uniform distribution on $\{0,1\}^\infty$.

Definition 2.23 (Martin-Löf test of randomness)**.** A *Martin-Löf test (of randomness)* is a computably enumerable set $U \subset \mathbb{N}\times\{0,1\}^*$ such that, with $U_n := \{x \in \{0,1\}^* : (n,x) \in U\}$, the condition $\lambda([\![U_n]\!]) \leq 2^{-n}$ is satisfied for all $n \in \mathbb{N}$. (We may identify the sequence $(U_n)_{n\in\mathbb{N}}$ with the test U.)
We say a sequence $\alpha \in \{0,1\}^\infty$ *passes* such a test, if $\alpha \notin \bigcap_{n\in\mathbb{N}} [\![U_n]\!]$.

Just for the sake of a better understanding we present the following link to Definition 2.22.

Theorem 2.24. *A sequence $\alpha \in \{0,1\}^\infty$ is Martin-Löf random if and only if α passes all Martin-Löf tests.*

We continue with introducing Levin's terminology. He works with a definition of tests (see his 1974 paper [Lev74]) that differs form the one by Martin-Löf in three aspects: First, we generalize to arbitrary computable measures. Second, we focus

on the maximum index n of the sets U_n in which a sequence is contained. And third and most important, we do *not implicitly require effectiveness*.

Definition 2.25 (Test)**.** Let P be a computable probability measure on $\{0,1\}^\infty$. A function $d : \{0,1\}^\infty \to \mathbb{N} \cup \{\infty\}$ is a *P-test* if

(i) $d(\alpha) = \max\{m : \alpha \in [\![x]\!] \text{ and } (m,x) \in U\}$, for some $U \subset \mathbb{N} \times \{0,1\}^*$;

(ii) $P(\{a \in \{0,1\}^\infty : d(a) \geq m\}) \leq 2^{-m}$, for all $m \geq 0$.

Now, mainly for the sake of completeness, we want to quote a different definition of tests that Levin gives in his newer works (e.g. [Lev10]).

Definition 2.26 (Integral test)**.** Let P be a recursive probability measure on $\{0,1\}^\infty$. A function $d : \{0,1\}^\infty \to \mathbb{R}$ is an *integral P-test*, if it satisfies the condition

$$\int_{\{0,1\}^\infty} 2^{d(\alpha)} P(d\alpha) \leq 1.$$

It should be mentioned, that every integral P-test is a P-test (in the sense of Definition 2.25). However, the converse does not hold true. We refer to Gacs [Gac] for further information on this topic. It seems that the independence conservation inequalities (Chapter 3) can be expressed using these tests, too, but we will stick to Definition 2.25.

2.3.3 Tests for finite sequences

The idea of tests for strings is similar to the one for infinite sequences. We will use such "finite" tests to express the independence conservation inequalities for strings in Section 3.1.2. The following notion of a sum test is a discrete analogon to the integral test (Definition 2.26). Note again, that in the present work we do not define tests as implicitly effective.

Definition 2.27 (Sum test)**.** Let P be a computable probability measure on $\{0,1\}^*$. A function $d : \{0,1\}^\infty \to \mathbb{R}$ is a *sum P-test* if it satisfies the condition

$$\sum_{x \in \{0,1\}^*} 2^{d(x)} P(x) \leq 1.$$

2.4 Church-Turing thesis and Gödel incompleteness

In this Section we basically want to discuss two issues. First, we introduce the Church-Turing thesis. Levin's independence postulate (Thesis 3.26) should be seen in analogy hereto; and the forbidden information thesis 4.14, which is more or less the central thesis of the present work, is an extension of Gödel's incompleteness theorem combined with the Church-Turing thesis. Second, we establish a tight link between consistent completions of Peano arithmetic and total extensions of the universal partial computable predicate. We use this link to present a reformulation of Gödel's incompleteness theorem; and to apply the forbidden information theorem 4.1 (which itself only makes an assertion with respect to total extensions of universal partial computable predicates) to completions of Peano arithmetic in Chapter 4.

At the end of this section, we will briefly introduce the notion of definability and the halting probability, which we will need in Chapter 4.

2.4.1 The Church-Turing thesis

Have in mind the definition of a computable function as a function computable by a Turing machine. The following thesis establishes a bridge between the informal concept "effective calculability" and the formal concept "computable function".[8]

[8]The notion of a Turing machine comprises two sides: the actual, "real-world" *machine* that we (may) imagine when we work with this notion and the formal machine that consists of the tape alphabet, transition function and so on. So the concept of the Turing machine itself already gives strong support for the Church-Turing thesis.

Thesis 2.28 (Church-Turing thesis)**.** Every effectively calculable function is a computable function.

Note that we consider the converse direction, i.e. that every computable function is effectively calculable, as implicit in the notion of "effective calculability".

Let us say a few words regarding the thesis. Obviously, since "effective calculability" is an informal expression, it needs an interpretation, and different interpretations are possible. The most important ones (pointed out by Gandy in his paper "Church's Thesis and Principles for Mechanisms" [Gan80], among others) are the following three.

(i) We consider a function as effectively calculable, if it is calculable by an abstract human being, using some mechanical aids.

(ii) We consider a function as effectively calculable, if it is calculable by a mechanical device.

(iii) We consider a function as effectively calculable, if it is calculable by any physically realizable device.

While Church and Turing essentially had interpretation (i) in mind[9], Gandy worked with interpretation (ii). Though Levin's independence postulate (Thesis 3.26) differs from the Church-Turing thesis, we will see that it is an assertion as least as "radical" as interpretation (iii) for the Church-Turing thesis.

2.4.2 Gödel's incompleteness theorem and the universal partial computable predicate

In this section, we restrict to first-order logic and the language of the natural numbers.

[9]See Gandy [Gan80], p. 1

One basic question in mathematical logic is, given some axiomatic system (i.e. a set of sentences): is it possible to find a *consistent, complete extension* of it? As already mentioned in the introduction, particularly this question was discussed with respect to the Peano axioms (for their precise definition, see Odifreddi [Odi92]). We denote their deductive closure, *Peano arithmetic*, by PA. By Lindenbaum's lemma, if PA is consistent (which we assume[10]), then there exists such an extension.[11] Still, Lindenbaum's lemma does not give us an *effectively constructible* extension, so the question remains open, whether such a constructible extension exists.

If we understand "effectively constructible" as "effectively calculable" - which implies that we only consider *deterministic* constructions - and accept the Church-Turing thesis, we can reformulate this question as whether there is a recursively axiomatizable consistent completion of PA. For this question we get a definite negative answer by Gödel's first incompleteness theorem. (Note that we drop the "first" form now on and only talk of "Gödel's incompleteness theorem".)

Have in mind that "recursively axiomatizable" means being the deductive closure of a recursive set of axioms; and obviously, if a set is recursively axiomatizable, it is necessarily recursively enumerable.

Theorem 2.29 (Incompleteness theorem)**.** *If T is a consistent, recursively enumerable extension of* PA*, then T is incomplete.*

Note that we will consider the combination of Gödel's incompleteness theorem and the Church-Turing thesis - i.e. that there is no effectively calculable consistent completion of PA - again in Chapter 4 and refer to it as "Gödel's thesis".

Without explicitly mentioning it, we used the fact that based on the well-known Gödel numbering, we can build a "structure preserving" bijection between $\mathbb{N}$ and

[10] The consistency of PA is proved to be not provable within PA itself by Gödel's second incompletness theorem. But within Zermelo-Fraenkel set theory, which we assume, it is in fact proved.

[11] The proof of Lindenbaum's lemma relies on the axiom of choice and is therefore quite unconstructive.

the sentences in the language of the natural numbers.[12] So we can identify sentences with elements of $\mathbb{N}$. Particularly, we can consider PA and any extension of it as subsets of $\mathbb{N}$.

Now we use this fact to make the following definition.

Definition 2.30 (PA-completeness)**.** A set $A \subset \mathbb{N}$ is called PA-*complete*, if one can compute relative to A a complete, consistent extension of PA.

We can reformulate the incompleteness theorem using PA-complete sets.

Theorem 2.31 (Incompleteness theorem, first reformulation)**.** *If a set A is* PA-*complete, then it is not computable.*

The equivalence of both formulations is immediate by trivial proofs by contradiction (note that any computably enumerable and complete theory is already computable).

Now we go one step further and express the notion of PA-completeness in terms of partial computable functions.

Definition 2.32. A partial computable function $f :\subseteq \{0,1\}^* \to \{0,1\}$ is called *universal partial computable predicate*, if for any other partial computable function $g :\subseteq \{0,1\}^* \to \{0,1\}$, there is some string p_g, such that

$$g(x) \cong f(p_g{}^\frown x), \text{ for all } x \in \{0,1\}^*.$$

We fix a universal partial computable predicate and denote it by $\mathbf{u}$.

Remark 2.33 (Existence of $\mathbf{u}$). Note that the existence of a universal partial computable predicate is guaranteed by the existence of a universal Turing machine.

Theorem 2.34. *Let $A \subset \mathbb{N}$. The following are equivalent:*

(i) A is PA-*complete.*

(ii) A computes a total extension of $\mathbf{u}$.

[12] By "structure preserving" we mean that from an $n \in \mathbb{N}$, we can exactly reconstruct the sentence it corrseponds to.

Theorem 2.34 is used by Levin [Lev10] and Stephan [Ste06], among others. However, we could not find a proof for it in the literature. The proof we now present for this theorem makes use of some results and proof ideas that we found in the monographs by Odifreddi [Odi92] and Downey and Hirschfeldt [DH10].

We will need the following lemma.

Lemma 2.35. *Let S be a set of sentences and φ be a sentence. Then $S \cup \{\neg\varphi\}$ is inconsistent, if and only if $S \vdash \varphi$.*

Proof of Theorem 2.34. **(i) $\Rightarrow$ (ii):** Let T be a consistent, complete extension of PA that is computable in A.

Since any partial computable function is representable in Peano arithmetic (for details see Odifreddi [Odi92]), there is a formula $\varphi(x)$ such that

$$\mathbf{u}(n)\downarrow = 1 \Leftrightarrow \mathrm{PA} \vdash \varphi(\tilde{n}),$$
$$\mathbf{u}(n)\downarrow = 0 \Leftrightarrow \mathrm{PA} \vdash \neg\varphi(\tilde{n}),$$

where $\tilde{n}$ denotes the numeral that represents n within the formal language of the natural numbers, i.e. is the n-th successor of the constant symbol 0, for all $n \in \mathbb{N}$.

We define for all $n \in \mathbb{N}$

$$\hat{u}(n) := \begin{cases} 1, & \text{if } \varphi(\tilde{n}) \in T, \\ 0, & \text{if } \neg\varphi(\tilde{n}) \in T. \end{cases}$$

Thus defined, $\hat{u}$ is an A-computable total extension of $\mathbf{u}$.

(ii) $\Rightarrow$ (i): Now let A be a set computing a total extension $\hat{u}$ of $\mathbf{u}$. We define a partial computable predicate P by $P := \varphi_M$ for the (partial) machine M that is constructed as follows. Remember that we identified $\mathbb{N}$ with the set of all sentences.

On input $\langle m, n\rangle$, M runs through all possible proofs and for each checks, if it proofs either n or $\neg n$ from $\mathrm{PA} \cup \{m\}$, or none. When a proof for n is found (before

one for $\neg n$), M stops and outputs 1; when a proof for $\neg n$ is found (before one for n), M stops and outputs 0.

Since P is partial computable, there is some string p, such that $Q(\cdot) :\cong \hat{u}(p^\frown\cdot)$ is a total extension of P.

Now we use this fact to define a set of sentences T, which is a complete, consistent extension of PA, computable in A. We do this by inductively and effectively defining finite sets $T_0 \subset T_1 \subset \ldots$ that are consistent with PA, and setting $T := \bigcup_{n\in\mathbb{N}} T_n$ - which is then, due to the compactness theorem, consistent with PA as well.

$\underline{n = 0:}$ We set $T_0 := \emptyset$.

$\underline{n \curvearrowright n+1:}$ Suppose we have defined the finite set T_n, that is consistent with PA. Let $m := \bigwedge_{k\in T_n} k$.

If $Q(\langle m, n\rangle) = 1$, then $P(\langle m, n\rangle) \ncong 0$. So either $\mathrm{PA} \cup \{m\} \vdash n$ or $P(\langle m, n\rangle)\uparrow$. In the first case, since $\mathrm{PA} \cup \{m\}$ is consistent, we know that $\mathrm{PA} \cup \{m\} \nvdash \neg n$. But in the latter case, by the construction of P, we have $\mathrm{PA} \cup \{m\} \nvdash \neg n$ as well. Hence, by setting $T_{n+1} := T_n \cup \{n\}$ we have T_{n+1} being consistent with PA (due to Lemma 2.35).

Otherwise, if $Q(\langle m, n\rangle) = 0$, then either $\mathrm{PA} \cup \{m\} \vdash \neg n$ or $P(\langle m, n\rangle)\uparrow$. By a similar argument as above, in both cases this means $\mathrm{PA} \cup \{m\} \nvdash n$. Setting $T_{n+1} := T_n \cup \{n\}$ we have T_{n+1} being consistent with PA (due to Lemma 2.35 again). ⌟

So by construction, T is a complete, consistent extension of PA, computable in A.

□

Note that we immediately get another reformulation of the incompleteness theorem.

Theorem 2.36 (Incompleteness theorem, second reformulation)**.** *If $\hat{u}$ is a total extension of* $\mathbf{u}$*, then $\hat{u}$ is not computable.*

The equivalence of this second reformulation and the first one (Theorem 2.31) is immediate by Theorem 2.34 and trivial proofs by contradiction.

2.4.3 Definability

There is one other subject from the field of mathematical logic that we want to treat now, namely definability of a set. We need to define definability in order to state the independence postulate (Thesis 3.26) later on. Note that for a structure M, we denote its domain by $\mathrm{dom}(M)$.

Definition 2.37 (Definable set)**.** Let L be a first-order language, M be an L-structure. A set $A \subset \mathrm{dom}(M)$ is called *definable in* M, if there exists an L-formula $\psi(x)$, such that

$$A = \{a \in \mathrm{dom}(M) : M \models \psi(a)\}.$$

There is a sequence definable in $\mathbb{N}$, i.e. the intended structure for the natural numbers, that is moreover a Martin-Löf random, left-c.e. real. It will be important for us in Chapter 4. Keep in mind that $\mathbf{U}$ denotes our universal prefix-free Turing machine.

Definition 2.38 (Halting probability)**.** The *halting probability* Ω is defined by

$$\Omega := \sum_{p \in \mathrm{dom}(\mathbf{U})} 2^{-p}.$$

For further details on this matter, we refer to Downey and Hirschfeldt [DH10], and Li and Vitanyi [LV08].

3 Mutual information

The concept of mutual information is the central instrument we will use to express and justify the forbidden information theorem 4.1 and the forbidden information thesis 4.14 in Chapter 4.

Interestingly, when Kolmogorov for the first time explicitly wrote about Kolomogorov complexity in his paper "Three approaches to the quantitative definition of information" [Kol68], he used Kolmogorov complexity mainly as a means to define mutual information. He considers the concept of "information conveyed by an object x about an object y" more fruitful than just "the information in an object x".[1]

How did Kolmogorov formalize the notion of mutual information? For finite sequences, his basic idea is to indirectly define mutual information using an algebraic expression: the quantity of information contained in a sequence x equals the quantity of information in x that is conveyed by y (i.e. their mutual information) plus the quantity of information in x, that is not conveyed by y. By letting $I(x : y)$ denote the mutual information of x and y, we may preliminarily(!) formalize this as

$$\mathbf{K}(x) = I(x : y) + \mathbf{K}(x|y),$$

or equivalently

$$I(x : y) = \mathbf{K}(x) - \mathbf{K}(x|y).$$

[1] The main reason he gives for this opinion is based upon a recourse to probabilistic information theory, where the entropy of a single continuous random variable X is often infinite but the mutual information with another random variable Y is finite and thus examinable. For futher information on probabilistic information theory we refer to MacKay [Mac03].

Observe that Kolmogorov made this definition[2] of mutual information in analogy to the one in probabilistic information theory, where we have

$$I(X : Y) = H(X) - H(X|Y)$$

for two random variables X, Y, with H denoting the (conditional) entropy.[3] (For futher information on this topic we refer to MacKay [Mac03].)

For several reasons, we will base our considerations on a definition of mutual information that slightly differs from the one given by Kolmogorov. However, Kolmogorov's definition is still widely used (parallely to the one we will consider) and gives a good intuition for the underlying idea of mutual information, so one should keep it in mind.

In this work we will discuss mutual information for both, finite and infinite sequences. The definition we use for strings is widely accepted. However, the definition we use for infinite sequences is due to Levin and almost exclusively used by him. Besides the definitions and some basic properties of mutual information, in this chapter the focus is on the independence conservation inequalities, which are due to Levin. They are central for the justification of the independence postulate, which we will present at the end of this chapter.

All definitions, theorems and proofs of the first two sections of this chapter are taken from the monographs by Downey and Hirschfeldt [DH10] and Li and Vitanyi [LV08], the lecture notes by Gacs [Gac] and Levin's papers [Lev74, Lev80, Lev84, Lev10]. Except for Theorem 3.17, Lemma 3.25, their proofs and some simple properties of mutual information, which are based on personal communication with Levin and own considerations.

[2]It should be mentioned that Kolmogorov actually used plain Kolmogorov complexity in his definition.

[3]Note that this expression is used for "infinite", i.e. continuous random variables, too.

3.1 Mutual information for finite sequences

We start with definitions and basic properties and afterwards show the independence consveration inequalities for finite sequences.

3.1.1 Definition and basic properties

As already mentioned, we define mutual information different for this work than in the introduction above. However, Theorem 3.2 will show that in fact both definitions are very similar.

Definition 3.1 (Mutual information for strings)**.** For $x, y \in \{0,1\}^*$, their *mutual information* $\mathbf{I}(x,y)$ is defined as

$$\mathbf{I}(x,y) := \mathbf{K}(x) + \mathbf{K}(y) - \mathbf{K}(x,y).$$

This definition seems intuitive as well: the difference of the information needed to describe strings x and y separately on the one hand and jointly on the other hand can be considered as their mutual information.

The main reason why we do not take the preliminary definiton of mutual information mentioned in the introduction, i.e. $\mathbf{K}(x) - \mathbf{K}(x|y)$, is, that this expression is not symmetric in the arguments x, y, as was shown by Zvonkin and Levin [ZL70]. Moreover, the independence conservation inequalties do not hold true for that expression (as Levin mentions without proof in "Forbidden Information"[Lev10]).

Theorem 3.2. *For all $x, y \in \{0,1\}^*$, we have*

$$\begin{aligned} \mathbf{I}(x,y) &\stackrel{+}{=} \mathbf{K}(x) - \mathbf{K}(x|y^*) \\ &\stackrel{+}{=} \mathbf{K}(y) - \mathbf{K}(y|x^*). \end{aligned}$$

This theorem is an immediate consequence of the following lemma.

Lemma 3.3 (Symmetry of information)**.** *For all $x, y \in \{0,1\}^*$ we have*

$$\mathbf{K}(x,y) \stackrel{+}{=} \mathbf{K}(x) + \mathbf{K}(y|x^*).$$

Proof. We first prove that

$$\mathbf{K}(x,y) \stackrel{+}{<} \mathbf{K}(x) + \mathbf{K}(y|x^*).$$

Let z be a minimal length description of y given x^*. Then we can construct a prefix-free machine M that, on input x^*z, first computes x from x^*, then computes y from x^* and z, and finally outputs $\langle x, y\rangle$.

In order to prove the reverse inequality, we prove that

$$\mathbf{K}(y|x^*) \stackrel{+}{<} \mathbf{K}(x,y) - \mathbf{K}(x),$$

using the KC theorem 2.14.

Let $z_1, z_2, \ldots$ be a recursive enumeration of $\mathrm{dom}(\mathbf{U})$ and let x_i, y_i be such that $\mathbf{U}(z_i) = \langle x_i, y_i\rangle$, for all i. Let $W_x := \{i : x_i = x\}$.

Given n and x, we build a KC set by enumerating requests $(l(z_i) - n, y_i)$ for $i \in W_x$, as long as the weight of these requests does not exceed 1. Call the resulting prefix-free machine $M_{n,x}$.

With Q being the "output probability" with respect to $\mathbf{U}$, as we defined it in Remark 2.17, there are constants c and c' such that

$$2^{\mathbf{K}(x)-c} \sum_{y\in\{0,1\}^*} Q(\langle x,y\rangle) \;\leq\; 2^{\mathbf{K}(x)-c'} Q(x) \;\leq\; 1.$$

The second inequality is due to the coding theorem 2.21. For the first inequality let us consider the Turing machine V, that outputs x, whenever $\mathbf{U}$ outputs $\langle x, y\rangle$ for some y. Since $x \mapsto Q_V(x) := \sum_{p:V(p)=x} 2^{-\ell(p)}$ is a lower semicomputable discrete

semimeasure, we have

$$Q(x) \stackrel{*}{>} Q_V(x) = \sum_{p:V(p)=x} 2^{-\ell(p)} = \sum_{y\in\{0,1\}^*} Q(\langle x,y\rangle),$$

by the coding theorem 2.21.

Since

$$\begin{aligned}\sum_{i\in W_x} 2^{-(l(z_i)-l(x^*)+c)} &= 2^{\mathbf{K}(x)-c} \sum_{y\in\{0,1\}^*} \sum_{z:\mathbf{U}(z)=\langle x,y\rangle} 2^{-l(z)} \\ &= 2^{\mathbf{K}(x)-c} \sum_{y\in\{0,1\}^*} Q(\langle x,y\rangle) \\ &\leq 1,\end{aligned}$$

all relevant requests will be enumerated by the Turing machine $M_{\mathbf{K}(x)-c,x}$.

Now we define the the oracle prefix-free machine M as follows. With w on the oracle tape, M first computes $x = \mathbf{U}(w)$ and then simulates $M_{l(w)-c,x}$.

If M has x^* on its oracle tape, then it will simulate $M_{\mathbf{K}(x)-c,x}$. Since there is an $i \in W_x$ such that $z_i = \langle x,y\rangle^*$, the KC set defining $M_{\mathbf{K}(x)-c,x}$ has as one of its requests $(\mathbf{K}(x,y) - \mathbf{K}(x) + c, y)$, thus

$$\mathbf{K}(y|x^*) \stackrel{+}{<} \mathbf{K}_M(y|x^*) \leq \mathbf{K}(x,y) - \mathbf{K}(x) + c.$$

□

In some cases we will need the following "conditional" version of mutual information, which may be seen in analogy to the conditional version of Kolmogorov complexity.

Definition 3.4. For $x,y,z \in \{0,1\}^*$ the *mutual information* $\mathbf{I}(x,y|z)$ *of* x,y *relative to* z is defined as

$$\mathbf{I}(x,y|z) := \mathbf{K}(x|z) + \mathbf{K}(y|z) - \mathbf{K}(x,y|z).$$

Again, we can reformulate this definition as follows.

Theorem 3.5. *For all $x, y, z \in \{0,1\}^*$, we have*

$$\begin{aligned}\mathbf{I}(x, y|z) &\stackrel{+}{=} \mathbf{K}(x|z) - \mathbf{K}(x|y, \mathbf{K}(y|z), z) \\ &\stackrel{+}{=} \mathbf{K}(y|z) - \mathbf{K}(y|x, \mathbf{K}(x|z), z).\end{aligned}$$

This theorem is an immediate consequence of the following lemma, that can be proved similarly to Lemma 3.3.

Lemma 3.6 (Symmetry of information, conditional version)**.** *We have for all $x, y, z \in \{0,1\}^*$*

$$\mathbf{K}(x, y|z) \stackrel{+}{=} \mathbf{K}(x|z) + \mathbf{K}(y|x, \mathbf{K}(x|z), z),$$

where the additive constant implicit in "$\stackrel{+}{=}$" does not depend on x, y, z.

We proceed with stating some basic properties of mutual information.

Theorem 3.7. *The mutual information* $\mathbf{I}$ *has the following properties:*

(i) $\mathbf{I}(x : y) \stackrel{+}{>} 0$*, for all $x, y \in \{0,1\}^*$.*

(ii) $\mathbf{I}$ *is symmetric, i. e.* $\mathbf{I}(x : y) = \mathbf{I}(y : x)$*, for all $x, y \in \{0,1\}^*$.*

(iii) $\mathbf{I}(x : y) \stackrel{+}{<} \min\{\mathbf{K}(x), \mathbf{K}(y)\}$*, for all $x, y \in \{0,1\}^*$.*

(iv) $\mathbf{I}(x : x) = \mathbf{K}(x)$*, for all $x \in \{0,1\}^*$.*

(v) $\langle x, y\rangle \mapsto \mathbf{I}(x : y)$ *is not lower semicomputable.*

Proof. **(i):** This follows from the fact that $\mathbf{K}(x) \stackrel{+}{>} \mathbf{K}(x|y^*)$.

(ii): This is obvious.

(iii): As $\mathbf{K}(y) \stackrel{+}{<} \mathbf{K}(x, y)$, we have $\mathbf{I}(x : y) \stackrel{+}{<} \mathbf{K}(x) + \mathbf{K}(y) - \mathbf{K}(y)$.

(iv): This equality is due to the fact that $\mathbf{K}(x) \stackrel{+}{=} \mathbf{K}(x, x)$

(v): If $\mathbf{I}$ was lower semicomputable, then $x \mapsto \mathbf{K}(x) \stackrel{+}{=} \mathbf{I}(x : x)$ would be lower semicomputable and thus computable. □

Intuitively, $\mathbf{I}(\langle x, y\rangle : z)$ should be greater than or equal to $\mathbf{I}(x : z)$. The following theorem implies that mutual information is in fact monotonic (which we state as a corollary).

Theorem 3.8. *For all $x, y, z \in \{0,1\}^*$ we have*

$$\mathbf{I}(\langle x, y\rangle : z) \stackrel{+}{=} \mathbf{I}(z : x) + \mathbf{I}(z : y|x^*).$$

Proof. We have

$$\begin{aligned}
\mathbf{I}(z : \langle x, y\rangle) - \mathbf{I}(z : x) &\stackrel{+}{=} \mathbf{K}(z) + \mathbf{K}(x, y) - \mathbf{K}(z, x, y) - \mathbf{K}(z) - \mathbf{K}(x) + \mathbf{K}(z, x) \\
&\stackrel{+}{=} \mathbf{K}(x, y) - \mathbf{K}(x) + \mathbf{K}(z, x) - \mathbf{K}(z, x, y) \\
&\stackrel{+}{=} \mathbf{K}(y|x^*) + \mathbf{K}(z|x^*) + \mathbf{K}(x) - \mathbf{K}(z, x, y) \\
&\stackrel{+}{=} \mathbf{K}(y|x^*) + \mathbf{K}(z|x^*) - \mathbf{K}(y, z|x^*) \\
&\stackrel{+}{=} \mathbf{I}(y : z|x^*),
\end{aligned}$$

where the third and fourth equalities are due to Theorem 3.3. □

Corollary 3.9. *The mutual information $\mathbf{I}$ is monotonic, i.e. for all $x, y, z \in \{0,1\}^*$ we have*

$$\mathbf{I}(x : \langle y, z\rangle) \stackrel{+}{>} \mathbf{I}(x : y).$$

3.1.2 Independence conservation inequalities

When Levin called the inequalities in Theorem 3.10 and 3.13 below "independence conservation inequalities"[4] he probably had in mind the famous laws from physics, such as the energy conservation law or the second law of thermodynamics which states that the entropy within a closed system can only increase.[5] The essence of the independence conservation inequalities is, that the information a string x

[4]The inequalties are called "independence conservation inequalities" in "Forbidden Information". In earlier papers Levin calls them "information conservation inequalities" [Lev74] or "randomness conservation inequalities" [Lev84].

[5]The second law of thermodynamics is not called a conservation law though.

conveys about a string y can not substantially be increased, neither by deterministic nor by randomized algorithmic processing of x. To state this differently, the independence of x and y is conserved, it cannot decrease. Note that we have not formally defined the term "independence", but we can roughly say that the independence of two strings increases when their mutual information decreases.

The first independence conservation inequality, which is about increase of mutual information by deterministic algorithms, is pretty obvious.

Theorem 3.10 (First independence conservation inequality for strings: mutual information non-increase by algorithms)**.** *Let $f\colon \{0,1\}^* \to \{0,1\}^*$ be a computable function. Then there is a constant c_f such that for all $x, y \in \{0,1\}^*$ we have*

$$\mathbf{I}(f(x) : y) \leq \mathbf{I}(x : y) + c_f.$$

Proof. We have

$$\mathbf{I}(x : y) = \mathbf{K}(x) + \mathbf{K}(y) - \mathbf{K}(x, y) \stackrel{+}{=} \mathbf{I}(\langle x, f(x)\rangle : y),$$

since from a description of x, we also get $\langle x, f(x)\rangle$ (by transforming $\mathbf{U}$ into a new machine that outputs $\langle \mathbf{U}(w), f(\mathbf{U}(w))\rangle$ on input w), and vice versa (by a similar machine), so $\mathbf{K}(x) \stackrel{+}{=} \mathbf{K}(x, f(x))$. And with a similar argument $\mathbf{K}(x, y) \stackrel{+}{=} \mathbf{K}(x, f(x), y)$. The additive error terms reflect the Turing machines we constructed from $\mathbf{U}$, which include an encoding of f.

But then by the monotonicity of $\mathbf{I}$ we get

$$\mathbf{I}(f(x) : y) \stackrel{+}{<} \mathbf{I}(\langle x, f(x)\rangle : y) \stackrel{+}{=} \mathbf{I}(x : y),$$

or, if we explicitly denote the sum of the additive constants implicit in "$\stackrel{+}{=}$" and "$\stackrel{+}{<}$" by c_f,

$$\mathbf{I}(f(x) : y) \leq \mathbf{I}(x : y) + c_f.$$

□

We turn to the second independence conservation inequality now, which concerns processing of information by randomized algorithms. Let us first prove the following lemmas, where the second one already contains the essence of the second conservation inequality.

Lemma 3.11. *For all $x, y, z \in \{0,1\}^*$ we have*

$$\mathbf{K}(x|y^*) \stackrel{+}{<} \mathbf{K}(x,z|y^*) \stackrel{+}{<} \mathbf{K}(x|z^*) + \mathbf{K}(z|y^*).$$

Proof. Using the monotony of $\mathbf{K}$ and Lemma 3.3 several times, we obtain

$$\begin{aligned}
\mathbf{K}(x|y^*) &\stackrel{+}{<} \mathbf{K}(x,z|y^*) \\
&\stackrel{+}{=} \mathbf{K}(x,y,z) - \mathbf{K}(y) \\
&\stackrel{+}{=} \mathbf{K}(x,y|z^*) + \mathbf{K}(z) - \mathbf{K}(y) \\
&\stackrel{+}{<} \mathbf{K}(x|z^*) + \mathbf{K}(y|z^*) + \mathbf{K}(z) - \mathbf{K}(y) \\
&\stackrel{+}{=} \mathbf{K}(y,z) + \mathbf{K}(x|z^*) - \mathbf{K}(y) \\
&\stackrel{+}{=} \mathbf{K}(z|y^*) + \mathbf{K}(x|z^*).
\end{aligned}$$

□

Lemma 3.12. *For all $x, y, z \in \{0,1\}^*$ we have*

$$\mathbf{m}(z|x^*) \cdot 2^{\mathbf{I}(z:y)-\mathbf{I}(x:y)} \stackrel{*}{<} \mathbf{m}(z|x^*, y, \mathbf{K}(y|x^*)),$$

where the multiplicative constant implicit in "$\stackrel{*}{<}$" does not depend on x, y, z.

Proof. We have

$$\begin{aligned}
\mathbf{I}(z:y) - \mathbf{I}(x:y) &= \mathbf{K}(y) - \mathbf{K}(y|z^*) - (\mathbf{K}(y) - \mathbf{K}(y|x^*)) \\
&= \mathbf{K}(y|x^*) - \mathbf{K}(y|z^*).
\end{aligned}$$

Hence

$$\begin{aligned} -\log\left(\mathbf{m}(z|x^*)\cdot 2^{\mathbf{I}(z:y)-\mathbf{I}(x:y)}\right) &= \mathbf{K}(z|x^*)+\mathbf{K}(y|z^*)-\mathbf{K}(y|x^*) \\ &\stackrel{+}{>} \mathbf{K}(y,z|x^*)-\mathbf{K}(y|x^*) \qquad \text{(Theorem 3.11)} \\ &\stackrel{+}{=} \mathbf{K}(z|x^*,y,\mathbf{K}(y|x^*)). \qquad \text{(Theorem 3.6)} \end{aligned}$$

□

Now we are ready to state the second independence conservation inequality for strings.

Theorem 3.13 (Second independence conservation inequality for strings: mutual information non-increase by randomized algorithms)**.** *Let $(P_x)_{x\in\mathbb{N}}$ be a family of uniformly lower semicomputable discrete measures. Then for all $x,y\in\{0,1\}^*$ we have*

$$\mathbf{I}(\langle x,z\rangle : y) \stackrel{+}{<} \mathbf{I}(x:y)+d_{x,y}(z), \qquad \text{for all } z\in\{0,1\}^*,$$

where $d_{x,y}$ is some sum P-test.

Proof. We simply show that

$$\tilde{d}_{x,y}(z) := \mathbf{I}(\langle x,z\rangle : y)-\mathbf{I}(x:y)$$

is a sum P_x-test (as defined in 2.27) up to an additive constant. We have

$$P_x(z) \stackrel{*}{<} \mathbf{m}(z|x) \stackrel{*}{<} \mathbf{m}(z|x^*) \stackrel{*}{<} \mathbf{m}(x,z|x^*),$$

for all x,z, by Remark 2.20, where the multiplicative constant implicit in the first

"$\overset{*}{<}$" depends on P of course. Hence

$$\begin{aligned}
\sum_{z\in\{0,1\}^*} P_x(z)\cdot 2^{\tilde{d}_{x,y}} &\overset{*}{<} \sum_{z\in\{0,1\}^*} \mathbf{m}(x,z|x^*)\cdot 2^{\mathbf{I}(\langle x,z\rangle:y)-\mathbf{I}(x:y)} \\
&\leq \sum_{z,w\in\{0,1\}^*} \mathbf{m}(w,z|x^*)\cdot 2^{\mathbf{I}(\langle w,z\rangle:y)-\mathbf{I}(x:y)} \\
&\leq \sum_{z,w\in\{0,1\}^*} \mathbf{m}(w,z|x^*,y,\mathbf{K}(y|x^*)) \qquad \text{(Lemma 3.12)} \\
&\overset{*}{<} 1.
\end{aligned}$$

We get a proper P_x-test $d_{x,y}$ that fulfills the claimed inequality by subtracting from $\tilde{d}_{x,y}$ the necessary constant. □

For an easier interpretation let us restate the theorem as follows. This version follows even more immediately from Theorem 3.12.

Corollary 3.14. *Let $(P_x)_{x\in\mathbb{N}}$ be a family of uniformly lower semicomputable discrete measures. Then for all $x,y\in\{0,1\}^*$ we have*

$$\sum_{z\in\{0,1\}^*} P_x(z)\cdot 2^{\mathbf{I}(z:y)-\mathbf{I}(x:y)} \leq c_P,$$

for some constant c_P.

Have in mind the discussion of randomized algorithms in Section 2.1.4. To each randomized algorithm, we have associated the computable discrete distribution $P_x(y)$ for the algorithm to output string y on input x. We apply the above corollary and obtain the following result.

Corollary 3.15. *The probability for a randomized algorithm to compute a string z with $\mathbf{I}(z:y)-\mathbf{I}(x:y)\geq m$ on input x, is less than or equal to 2^{-m}, for any y, up to a multiplicative constant depending on the algorithm.*

3.2 Mutual information for infinite sequences

The concept of mutual information for infinite sequences is a pretty interesting subject, some nice applications can be found in Levin's 1984 paper [Lev84]. The important assertion we will make in Chapter 4, that, under the assumption of the independence postulate, a completion of Peano arithmetic is unrealistic, is another notable application. However, as already mentioned, there is no consensus on a precise definition of mutual information for infinite sequences. Levin alone mentions four varying definitions in three papers [Lev74, Lev80, Lev84].

We will base our considerations on the main definition that Levin proposes in his 1974 paper, although some counterintuitive properties of this definition have been proven recently (see Remark 3.20 below). Similar to the case of finite sequences, we will state two independence conservation inequalities.

3.2.1 Definition and basic properties

The definition of mutual information for infinite sequences is based on the universal discrete semimeasure $\mathbf{m}$ and the definition of mutual information for strings. Note that it includes infinite and finite sequences, we will show soon, that it is a proper generalization of the definition we gave in the previous section.

Definition 3.16 (Mutual information for infinite sequences)**.** For $a, b \in \{0,1\}^* \cup \{0,1\}^\infty$, their *mutual information* $\hat{\mathbf{I}}(a:b)$ is defined as

$$\hat{\mathbf{I}}(a:b) := \log \sum_{x,y \in \{0,1\}^*} \mathbf{m}(x|a) \cdot \mathbf{m}(y|b) \cdot 2^{\mathbf{I}(x:y)}.$$

If we neglect, that we switch between the linear and the logaritmic scale within the definition, we may think of this definition as follows: We calculate the weighted average of mutual information for all pairs of strings, where for each pair (x, y) the weight we give its mutual information is the "probability" that x is computed by a, and y is computed by b, respectively. Note that, since $\mathbf{m}(x|a) = 2^{-\mathbf{K}(x|a)}$, the

"probability" $\mathbf{m}(x|a)$ is great, if $\mathbf{K}(x|a)$ is small, i.e. if a contains much information about x.

It may also help to imagine (keeping Kolmogorovs definition from the introduction of this chapter in mind) that if

$$[\mathbf{K}(x) - \mathbf{K}(x|a)] + [\mathbf{K}(y) - \mathbf{K}(y|b)] - \mathbf{K}(x,y) \gg 0,$$

i.e. if the information, that a conveys about x plus the information, that b conveys about y is greater than the information contained in x and y together - then the information in a (about x) and the information in b (about y) necessarily have to "overlap".

We proceed with showing some basic properties of mutual information for infinite sequences. As already indicated, we have equivalence to the finite version of mutual information in the case of strings.

Theorem 3.17 (Equivalence in the finite case)**.** *For all $u, v \in \{0,1\}^*$ the two versions of mutual information coincide, i.e.*

$$\mathbf{I}(u:v) \stackrel{+}{=} \hat{\mathbf{I}}(u:v).$$

Proof. We have to show that

$$2^{\mathbf{I}(u:v)} \stackrel{*}{=} \sum_{x,y \in \{0,1\}^*} \mathbf{m}(x|u) \cdot \mathbf{m}(x|v) \cdot 2^{\mathbf{I}(x:y)}.$$

"$\stackrel{*}{<}$": This inequality is immediate, since the terms on the right-hand side are all positive and for $x = u$ and $y = v$, the left-hand side appears in the summation.

"$\stackrel{*}{>}$": To prove this inequality, we simply need to apply Corollary 3.14 twice (note that $\mathbf{m}(x|y) \stackrel{*}{<} \mathbf{m}(x|y^*)$, for any x, y, as $x^* \mapsto x$ is computable by a prefix-free

Turing machine):

$$\begin{aligned}
&\sum_{x,y\in\{0,1\}^*} \mathbf{m}(x|u)\cdot\mathbf{m}(y|v)\cdot 2^{\mathbf{I}(x:y)}\\
&= \sum_{x\in\{0,1\}^*} \mathbf{m}(x|u)\cdot 2^{\mathbf{I}(x:v)}\left[\sum_{y\in\{0,1\}^*} \mathbf{m}(y|v)\cdot 2^{\mathbf{I}(x:y)-\mathbf{I}(x:v)}\right]\\
&\overset{*}{<} \sum_{x\in\{0,1\}^*} \mathbf{m}(x|u)\cdot 2^{\mathbf{I}(x:v)-\mathbf{I}(u:v)}\cdot 2^{\mathbf{I}(u:v)}\\
&\overset{*}{<} 2^{\mathbf{I}(u:v)}.
\end{aligned}$$

Note that the multiplicative constant that is mentioned in Corollary 3.14 in the case at hand only depends on $\mathbf{m}$.

□

By a similar proof as for Theorem 3.17 we obtain the following result.

Corollary 3.18. *For all $\alpha\in\{0,1\}^\infty$, $x\in\{0,1\}^*$, we have*

$$\hat{\mathbf{I}}(\alpha:x)\overset{+}{=}\log\sum_{w\in\{0,1\}^*}\mathbf{m}(w|\alpha)2^{\mathbf{I}(w:x)}.$$

We proceed with listing some further properties of $\hat{\mathbf{I}}$.

Theorem 3.19. *The mutual information $\hat{\mathbf{I}}$ has the following properties:*

(i) $\hat{\mathbf{I}}(\alpha:x)\overset{+}{<}\mathbf{K}(x)$*, for all* $\alpha\in\{0,1\}^\infty, x\in\{0,1\}^*$.

(ii) If $\alpha\in\{0,1\}^\infty$ *is computable, then there is some constant* c_α*, such that* $\hat{\mathbf{I}}(\alpha:x)\overset{+}{<}c_\alpha$*, for all* $x\in\{0,1\}^*$.

(iii) Let $\alpha\in\{0,1\}^\infty$*, such that there is some constant* c*, so that* $\mathbf{K}(\alpha\restriction n)\geq 2\,\mathbf{K}(n)-c$*, for all* n *(e.g.* α *is Martin-Löf random). Then* $\hat{\mathbf{I}}(\alpha:\alpha)=\infty$.

(iv) $\hat{\mathbf{I}}$ *is monotonic, i.e.* $\hat{\mathbf{I}}(\alpha:b)\overset{+}{<}\hat{\mathbf{I}}(\langle\alpha,\gamma\rangle:b)$*, for all* $\alpha,\gamma\in\{0,1\}^\infty, b\in\{0,1\}^*$.

(v) Let $\alpha, \beta \in \{0,1\}^\infty$, such that α is Martin-Löf random and computable in β. Then $\hat{\mathbf{I}}(\alpha : \beta) = \infty$.

Proof. **(i):** We have

$$\begin{aligned} 2^{\hat{\mathbf{I}}(\alpha:x)} &\stackrel{*}{=} \sum_{w\in\{0,1\}^*} \mathbf{m}(w|\alpha) 2^{\mathbf{I}(w:x)} && \text{(Corollary 3.18)} \\ &\stackrel{*}{<} 2^{\mathbf{K}(x)} \sum_{w\in\{0,1\}^*} \mathbf{m}(w|\alpha) && \text{(Theorem 3.7)} \\ &\leq 2^{\mathbf{K}(x)}. \end{aligned}$$

(ii): As α is computable, there is a constant c_α, such that $\mathbf{m}(x|\alpha) \leq c_\alpha \, \mathbf{m}(x|\epsilon)$, for all x. Hence

$$\begin{aligned} 2^{\hat{\mathbf{I}}(\alpha:x)} &\stackrel{*}{=} \sum_{w\in\{0,1\}^*} \mathbf{m}(w|\alpha) 2^{\mathbf{I}(x:w)} && \text{(Corollary 3.18)} \\ &\stackrel{*}{<} c_\alpha \sum_{w\in\{0,1\}^*} \mathbf{m}(w|\epsilon) 2^{\mathbf{I}(x:w)-\mathbf{I}(x:\epsilon)} && \text{(Theorem 3.7, part (iii))} \\ &\stackrel{*}{<} c_\alpha. && \text{(Corollary 3.14)} \end{aligned}$$

(iii): We have

$$\begin{aligned} 2^{\hat{\mathbf{I}}(\alpha:\alpha)} &\stackrel{*}{>} \sum_{n\in\mathbb{N}} \mathbf{m}(\alpha\restriction n|\alpha)\, \mathbf{m}(\alpha\restriction n|\alpha) 2^{\mathbf{I}(\alpha\restriction n:\alpha\restriction n)} \\ &\stackrel{*}{>} \sum_{n\in\mathbb{N}} 2^{-2\,\mathbf{K}(n)+\mathbf{K}(\alpha\restriction n)} \\ &\stackrel{*}{>} \sum_{n\in\mathbb{N}} 2^{-c} \\ &= \infty, \end{aligned}$$

where the second inequality holds true due to Theorem 3.7 and the fact that $\mathbf{K}(\alpha\restriction n|\alpha) \stackrel{+}{<} \mathbf{K}(n)$, for all n.

(iv): The monotonicity follows immediately from the definition.

(v): Obviously, $\hat{\mathbf{I}}(\alpha : \alpha) \leq \hat{\mathbf{I}}(\alpha : \beta) + c$, for some constant c (for a formal proof, see Theorem 3.21 below). Then apply (iii). □

Remark 3.20. We end this section by mentioning some problematic aspect about $\hat{\mathbf{I}}$ that was found by Hirschfeldt and Weber [HW12]. As usual, we call a sequence α $\mathbf{K}$-*trivial*, if $\mathbf{K}(\alpha\restriction n) \leq \mathbf{K}(n)+c$, for all n and a constant c that does not depend on n. Furthermore, we say a sequence α has *finite self-information*, if $I(\alpha : \alpha) < \infty$, for whatever version of mutual information I we take.
It was proposed, also by Levin, that with an intuitive definition of mutual information I for infinite sequences, a sequence α should have finite self-information if and only if α is $\mathbf{K}$-trivial, i.e. for both notions the classes of the least complex sequences should coincide. But considering $I = \hat{\mathbf{I}}$, though $\mathbf{K}$-triviality implies finite self-information, finite self-information *does not* imply $\mathbf{K}$-triviality, as was shown by Hirschfeldt and Weber. This may be seen as an argument, that our version $\hat{\mathbf{I}}$ of mutual information has some deficiencies.

3.2.2 Independence conservation inequalities

The two independence conservation inequalities for infinite sequences are similar to the ones for finite sequences (see Section 3.1.2). The first inequality concerns algorithmic operators. Remember that an algorithmic operator is a function $f\colon \{0,1\}^\infty \to \{0,1\}^\infty$, such that there is an oracle Turing machine M with $f = \Phi_M$.

Theorem 3.21 (First independence conservation inequality for infinite sequences: mutual information non-growth by algorithmic operators)**.** *Let f be an algorithmic operator. Then there is a constant c_f, such that for all $\alpha, \beta \in \{0,1\}^\infty$ we have*

$$\hat{\mathbf{I}}(f(\alpha) : \beta) \leq \hat{\mathbf{I}}(\alpha : \beta) + c_f.$$

The observation is pretty obvious but we give a formal proof anyway.

Proof. Let M be an oracle Turing machine that computes f in the above sense. Let T be the oracle machine that works like the universal prefix-free machine $\mathbf{U}$, except that, given oracle α, whenever $\mathbf{U}^\alpha$ works with the bit $\alpha(i)$, T works with $M^\alpha(i)$ instead (by simulating M). Then $\mathbf{U}^{f(\alpha)}(x) = T^\alpha(x)$ and hence

$$\mathbf{K}(x|f(\alpha)) = \mathbf{K}_T(x|\alpha) \geq \mathbf{K}(x|\alpha) - c_T,$$

for some constant c_T (by Theorem 2.9). We define $c_f := c_T$ and obtain

$$\begin{aligned}\hat{\mathbf{I}}(f(\alpha) : \beta) &= \log \sum_{x,y\in\{0,1\}^*} 2^{-\mathbf{K}(x|f(\alpha))} \cdot \mathbf{m}(x|\beta) \cdot 2^{\mathbf{I}(x:y)} \\ &\leq \hat{\mathbf{I}}(\alpha : \beta) + c_f.\end{aligned}$$

□

Now we turn to the second conservation inequality for infinite sequences. As was the case with the second conservation inequality for strings, Theorem 3.13, this probabilistic statement gets particularly interesting when we apply it to randomized operators which we introduced in Section 2.1.4: to each finite composition of randomized operators, we have associated the α-computable probability $P_\alpha([\![y]\!])$ for the composition of operators to compute a sequence starting with the prefix y, on input α. The statement shows, that the probability of a substantial increase of mutual information by a compositions of randomized operators, and thus also by a single randomized operator, is vanishingly small.

Theorem 3.22 (Second independence conservation inequality for infinite sequences: mutual information non-growth by randomized operators)**.** *Let $(P_\alpha)_{\alpha\in\{0,1\}^\infty}$ be a family of uniformly α-computable continuous probability measures. Then for all $\alpha, \beta \in \{0,1\}^\infty$ we have*

$$\hat{\mathbf{I}}(\langle\alpha, \gamma\rangle : \beta) \leq \hat{\mathbf{I}}(\alpha : \beta) + d_{\alpha,\beta}(\gamma) + c_{\alpha,\beta}, \quad \textit{for all } \gamma \in \{0,1\}^\infty,$$

where $d_{\alpha,\beta}$ is some sequential P_α-test and $c_{\alpha,\beta}$ is some constant.

Keeping the notations of the theorem, it immediately follows from the theorem

and the definition of tests (Definition 2.25) that

$$\begin{aligned}
&P_\alpha(\{\gamma \in \{0,1\}^\infty : \hat{\mathbf{I}}(\langle \alpha, \gamma\rangle : \beta) - \hat{\mathbf{I}}(\alpha : \beta) > m\}) \\
&\leq P_\alpha(\{\gamma \in \{0,1\}^\infty : d_{\alpha,\beta}(\gamma) > m - c_{\alpha,\beta}\}) \\
&\leq 2^{-m+c_{\alpha,\beta}}.
\end{aligned}$$

So for an easier interpretability of the theorem, we state the following corollaries

Corollary 3.23. *Let $(P_\alpha)_{\alpha\in\{0,1\}^\infty}$ be a family of uniformly α-computable continuous probability measures. Then for all $\alpha, \beta \in \{0,1\}^\infty$ we have*

$$P_\alpha(\{\gamma \in \{0,1\}^\infty : \hat{\mathbf{I}}(\langle \alpha, \gamma\rangle : \beta) - \hat{\mathbf{I}}(\alpha : \beta) > m\}) \leq 2^{-m+c_{\alpha,\beta}},$$

for some constant $c_{\alpha,\beta}$.

Corollary 3.24. *The probability for a composition of randomized operators to compute a sequence γ with $\hat{\mathbf{I}}(\langle \alpha, \gamma\rangle : \beta) - \hat{\mathbf{I}}(\alpha : \beta) > m$ on input α, is less than or equal to 2^{-m}, for any β, up to a multiplicative constant depending on the algorithm and α, β.*

Unfortunately, we are not able to give a proof for Theorem 3.22. In his 1974 paper [Lev74], where Levin states the theorem, he gives no proof and not even any hint how to prove it.

We were only able to prove the following lemma based on some standard methods. We state it since we suppose that the proof idea can be taken as a basis to proof Theorem 3.22.

Lemma 3.25. *Let $(P_\alpha)_{\alpha\in\{0,1\}^\infty}$ be a family of uniformly α-computable continuous probability measures. Then for all $\alpha \in \{0,1\}^\infty$, $x \in \{0,1\}^*$ we have*

$$P(\{\gamma \in \{0,1\}^\infty : \mathbf{K}(x|\alpha,\gamma) \leq \mathbf{K}(x|\alpha) - c\}) < 2^{-c+d},$$

for all $c \in \mathbb{N}$ and some constant d.

Proof. Let α be arbitrary but fixed. Let S_x be the set of all pairs $(g,p) \in \{0,1\}^* \times \{0,1\}^*$, such that $\mathbf{U}^{\langle \alpha, g0\ldots\rangle}(p)$ has exactly use $2\,\ell(g)$ or $2\,\ell(g)+1$ on the oracle tape (i.e. reads g and the corresponding initial segment of α) and outputs x.

We build an oracle machine M that enumerates a KC set in the following way. M^α internally enumerates S_x for all $x = 0, 1, \ldots$. Whenever for the first time, for some x and c and the finite subset S'_x of S_x enumerated so far,

$$\sum_{(g,p)\in S'_x} 2^{-\,\ell(p)} \cdot P_\alpha([\![g]\!]) \;\geq\; 2^{-c},$$

M^α enumerates the KC request $\langle c+1, x\rangle$.

We have to prove that the requests enumerated by M form in fact a KC set. Let $\langle c_0+1, x_0\rangle, \langle c_1+1, x_1\rangle, \ldots$ denote these requests. We have to show that $\sum_{i\in\mathbb{N}} 2^{-c_i-1} \leq 1$.

Let S denote the set of all pairs (g,p), such that $\mathbf{U}^{\langle \alpha, g0\ldots\rangle}(p)$ has exactly use $2\,\ell(g)$ or $2\,\ell(g)+1$ on the oracle tape and terminates (i.e. $S = \bigcup_{x\in\{0,1\}^*} S_x$). Then

$$\begin{aligned}
\sum_{i\in\mathbb{N}} 2^{-c_i-1} &= \sum_{x\in\{0,1\}^*} 2^{-1} \sum_{i:\, x_i = x} 2^{-c_i} \\
&\leq \sum_{x\in\{0,1\}^*} 2^{-1} \sum_{k\in\mathbb{N}} 2^{-k} \sum_{(g,p)\in S_x} 2^{-\,\ell(p)} \cdot P_\alpha([\![g]\!]) \\
&= \sum_{(g,p)\in S} 2^{-\,\ell(p)} \cdot P_\alpha([\![g]\!]) \\
&= \sum_{(g,p)\in S} P_\alpha \otimes \lambda([\![g]\!] \times [\![p]\!]) \\
&= P_\alpha \otimes \lambda \left(\bigcup_{(g,p)\in S} [\![g]\!] \times [\![p]\!] \right) \leq 1,
\end{aligned}$$

with λ being the uniform distribution on $\{0,1\}^\infty$. The last equality holds true, since for all $(g,p), (g',p') \in S$, if

$$[\![g]\!] \times [\![p]\!] \cap [\![g']\!] \times [\![p']\!] \neq \emptyset,$$

then g is a prefix of g' or vice versa; and p is a prefix of p' or vice versa. But then $g = g'$ and $p = p'$, as otherwise there has to be some point of time, where $\mathbf{U}^{\langle\alpha,g0\ldots\rangle}(p)$ for the first time behaves different from $\mathbf{U}^{\langle\alpha,g'0\ldots\rangle}(p')$ which yields a contradiction (since up to that point of time, they read the same inputs).

So M^α enumerates a KC set. By the KC theorem 2.14 there is an oracle machine M' so that by Theorem 2.9 there is a constant d' such that

$$\mathbf{K}(x|\alpha) \leq \mathbf{K}_{M'}(x|\alpha) + d',$$

for all x. Let $d := d' + 2$. Then

$$P(\underbrace{\{\gamma \in \{0,1\}^\infty : \ \mathbf{K}(x|\alpha,\gamma) \leq \mathbf{K}(x|\alpha) - c\}}_{=:T_x}) < 2^{-c+d},$$

for all x and c.

Assume otherwise; then there is some x and c, such that $P_\alpha(T_x) \geq 2^{-c+d}$. Note that $T_x = [\![T'_x]\!]$, with T'_x denoting the set of all g, such that there is a p with $\ell(p) \leq \mathbf{K}(x|\alpha) - c$, and $\mathbf{U}^{\langle\alpha,g0\ldots\rangle}(p)$ has exactly use $2\,\ell(g)$ or $2\,\ell(g)+1$ on the oracle tape and outputs x.

We then have

$$\begin{aligned}\sum_{(g,p)\in S_x} 2^{-\,\ell(p)} \cdot P_\alpha([\![g]\!]) \ &\geq \ \sum_{g\in T'_x} 2^{-\,\mathbf{K}(x|\alpha)+c} \cdot P_\alpha([\![g]\!]) \\ &\geq 2^{-\,\mathbf{K}(x|\alpha)+c} \cdot P(T_x) \\ &\geq 2^{-\,\mathbf{K}(x|\alpha)+d}.\end{aligned}$$

Therefore, M^α enumerates a KC request $\langle \mathbf{K}(x|\alpha) - d + 1, x\rangle$, and hence

$$\begin{aligned}\mathbf{K}(x|\alpha) &\leq \mathbf{K}_{M'}(x|\alpha) + d - 2 \\ &\leq [\mathbf{K}(x|\alpha) - d + 1] + d - 2 \\ &= \mathbf{K}(x|\alpha) - 1,\end{aligned}$$

which is a contradiction.

□

3.3 The independence postulate

In this section we want to discuss Levin's independence postulate. It is a non-mathematical statement which we will need to argue for the non-mathematical forbidden information thesis 4.14 in Chapter 4. After stating the postulate, we will first try to justify it using the independence conservation inequalities and afterwards take a critical point of view.

3.3.1 The postulate

Though being quite interesting, Levin's independence postulate is a pretty difficult issue, which can already be seen by the fact, that he formulated four different versions in the four papers where it plays a role [Lev74, Lev80, Lev84, Lev10]. We will give an own formulation below. Though it slightly differs from all versions stated by Levin, it is more or less a logical consequence of all these versions. In particular, it is a comparatively weak formulation, but nevertheless it is strong enough for our purposes. For the reason why we do not use the version in "Forbidden Information", see Section 3.3.3, point (iv).

Thesis 3.26 (Independence postulate)**.** Let α be an infinite sequence that is definable in $\mathbb{N}$.[6] Then for an infinite sequence β that is generated by any locatable physical process, we have $\hat{\mathbf{I}}(\alpha : \beta) < \infty$.

Obviously, the independence postulate is not a mathematical statement, as it uses the non-mathematical term "infinite sequence that is generated by any locatable

[6] The reason to specify α this way is mainly, that we need to fix α somehow before considering empirical sequences. If we left α arbitrary, it could be the result of a locatable physical process (see [Lev74], p. 208).

physical process". It should be seen in analogy to the Church-Turing thesis 2.28, which contains the non-mathematical term "effective calculability".[7]

3.3.2 An attempt of justification

Now we want to present an argumentation supporting the independence postulate. It bases on ideas that Levin vaguely indicates in three of his papers [Lev80, Lev84, Lev10] but is far more elaborate. Although the basic ideas are rather simple, we give a detailed formulation to make its strengths and weaknesses better visible. Before beginning with the actual argumentation, let us make the following observations:

(i) An algorithmic operator $f\colon \{0,1\}^\infty \to \{0,1\}^\infty$ can always be regarded as a randomized operator, too (we can easily construct a randomized operator (M, Q) that calculates f with probability one). Particularly, we can regard a composition of both, randomized and algorithmic operators as a composition of only randomized operators. So by our argumentation in Section 2.1.4, the probability $P_\gamma(\llbracket z \rrbracket)$ for a composition of algorithmic and randomized operators to output a sequence with the initial segment z, on input γ, is γ-computable.

(ii) In the below argumentation we will have a family of measures $(P^i_{\beta_i^0})_{i\in I}$. Due to basic measure theory, we can define the product measure $R := \otimes_{i\in I} P^i_{\beta_i^0}$ on (the product σ-algebra on) $\prod_{i\in I}\{0,1\}^\infty$. R is the unique measure that

[7] It should be mentioned that the independence postulate and the Church-Turing thesis are not directly comparable in terms of logical implication. The Church-Turing thesis does definitely not imply the independence postulate, since it only makes assertion with respect to effectively calculable sequences and not sequences generated by locatable physical processes. *Regarding the reverse direction*, we may admittedly say that each effectively calculable sequence may be generated by some locatable physical process. However, since we restrict to the countable language of $\mathbb{N}$, there are only countably many uniquely definable α's and for each such α, by the independence conservation inequality, the set of sequences that has infinite mutual information with α has Lebesgue measure zero. So the set of sequences the independence postulate rules out to be effectively calculable has measure zero. But the set of non-computable sequences has measure one.

preserves the single measures on "their" spaces and makes events in different spaces independent.

The basic idea for the argumentation for the independence postulate is that any physical process can be modeled by a composition of randomized and algorithmic operators. And such a composition can produce infinite information about a previously defined sequence only with probability zero, due to the conservation inequalities. More formally:

(1) We postulate that there is an index set I, a family of compositions of algorithmic and randomized operators, $(C_i)_{i\in I}$, and a family of initial sequences, $(\beta_i^0)_{i\in I}$, such that the following holds true: For each sequence β generated by any locatable physical process there is an $i \in I$ such that β is the result of C_i applied to an initial sequence β_i^0.

(2) We postulate that for any sequence α that is definable in $\mathbb{N}$, and for all $i \in I$, we have $\hat{\mathbf{I}}(\alpha : \beta_i^0) < \infty$.

(3) Furthermore, we postulate that I is countable.

(4) Then the probability $P^i_{\beta_i^0}([\![z]\!])$ for the composition of operators C_i to output a sequence with the initial segment z, on input β_i^0, is β_i^0-computable. Applying Corollary 3.23 (and using the monotonicity of $\hat{\mathbf{I}}$) we obtain

$$P^i_{\beta_i^0}\Big(\{\beta : \hat{\mathbf{I}}(\alpha : \beta) > m\}\Big) \leq 2^{-m-\hat{\mathbf{I}}(\alpha:\beta_i^0)+c_{\beta_i^0,\alpha}}, \quad \text{for all } m \in \mathbb{N},$$

for some constant $c_{\beta_i^0,\alpha}$, and therefore

$$P^i_{\beta_i^0}\Big(\{\beta : \hat{\mathbf{I}}(\alpha : \beta) = \infty\}\Big) = P_{\beta_i^0}\left(\bigcap_{m\in\mathbb{N}} \{\beta : \hat{\mathbf{I}}(\alpha : \beta) > m\}\right) = 0,$$

for each $i \in I$.

(5) For a fixed $i \in I$, we have for the product measure R

$$R(\ C_i \text{ outputs } \beta \text{ with } \hat{\mathbf{I}}(\alpha : \beta) = \infty, \text{ on input } \beta_i^0\ )$$
$$= P^i_{\beta_i^0}\Big(\{\beta : \hat{\mathbf{I}}(\alpha : \beta) = \infty\}\Big) = 0.$$

Therefore

$$R(\text{ there is an } i \in I, \text{ so that } C_i \text{ outputs } \beta \text{ with } \mathbf{I}(\alpha : \beta) = \infty, \text{ on input } \beta_i^0\ )$$
$$\leq \sum_{i \in I} R(\ C_i \text{ outputs } \beta \text{ with } \hat{\mathbf{I}}(\alpha : \beta) = \infty, \text{ on input } \beta_i^0\ ) = 0.$$

The latter statement essentially coincides with the independence postulate we stated above.

3.3.3 A critical discussion

We proceed with taking a critical view on the independence postulate and its justification. The following points should be mentioned.

(i) With respect to step (1): the postulate would cover the following physical model of the world. We think of the β's as states of locatable parts of the universe. We represent the time evolution of the state of some part i of the universe as a sequence $\beta_i^0, \beta_i^1, \ldots$ consisting of sequences β_i^k in the state space $\{0,1\}^\infty$. Particularly, we have $\beta = \beta_i^{\hat{k}}$ for some $\hat{k}$. We assume that each transition from β_i^k to β_i^{k+1} is either a deterministic or a random transformation, which we represent by an algorithmic or randomized operator, respectively.

Although this physical model contains important aspects of classical mechanics and even grasps an essential aspects of quantum mechanics, namely the random character of measuerements, it is rather weak. Continuity seems an essential constituent of reality. But for our argumentation to work, we have to limit the growth of information by restricting to discrete evolution and moreover only considering a finite number of steps. Moreover, there seems to be

no proper reason, why the probability distributions of the random transformations sould necessarily be *computable* - but we model them by randomized operators that (by our definition) have computable distributions.

It needs to be mentioned, that the above physical model is not the only model with respect to which we may interpret the independence postulate. We could also think of the β's to be *infinite in time*. We then imagine them to be succesively written down by a finite composition of (randomized) Turing machines. In this case too, our above argumentation in favor of the independence postulate is applicable.

(ii) Regarding step (2) we can say, that this postulate is made up out of thin air. We could support it (based on the independence conservation inequalty 3.22) if we assumed the initial β_i^0's to be distributed by computable distributions. But for what reasons should we think that, for example, the initial state of the universe was "chosen" according to a computable distribution?

(iii) To support the postulate in step (3), we emphazise that we only talk about sequences β, that are generated by *locatable* physical processes. With our current language it seems that we can only refer to a countable number of entities, if we want to have distinct references for distinct entities. We assume that if a physical entity is locatable, it has a unique (spatiotemporal) reference. So there are only countably many locatable physical processes. (This is the reason why we do not simply talk about arbitrary physical processes.)

(iv) Why did we not take the version of the independence postulate, that Levin states in his paper "Forbidden Information"? Let us cite this version (see [Lev10], p. 7):

> "Let X be a sequence defined with an n-bit mathematical statement (e.g., in PA or set theory). Suppose a sequence Y can be located in the physical world with a k-bit instruction set. Then $\hat{\mathbf{I}}(X : Y) < k + n + c$, for some small absolute constant c."

We did not take this formulation since it is ambiguous and seems very difficult to justify. Particularly, it is pretty unclear what is meant by the locatability

"with a k-bit instruction set". If we can chose the instruction set after having full information about the sequence Y, then we may chose it in a way that the inequality $\hat{\mathbf{I}}(X:Y) < k+n+c$ is fulfilled. However, if we do not completely know the sequence Y when we chose the instruction set by which we locate it, then for what reasons should the inequality $\hat{\mathbf{I}}(X:Y) < k+n+c$ be fulfilled?

Another point is the following. When we talk about a sequence that is generated by any locatable physical process, then we have a more or less well described and graspable entity. We modeled a physical process by a composition of randomized and algorithmic operators and this way we were able to apply the independence conservation inequalites. Compared with this, "a sequence that can be located in the physical world" seems a pretty vague expression and it is difficult to justify any assertion about it. (In some sense, any sequence can be located in the physical world, if we assume the sequence $0101\ldots$ to be locatable.)

The following should be mentioned, too. It seems that based on the independence conservation inequalities, we can only make assertions that have a somehow probabilistic form. The probabilistic form of our version of the independence postulate is hidden but gets obvious by the argumentation we gave for it: it holds with probability one. But it seems that the version of the independence postulate in "Forbidden Information" cited above has no probabilistic form at all. Note that Levin in his 1974 paper [Lev74] states one draft version that has a probabilistic form, and one version that is somehow probabilistic, too, by having a frequentist form.

To *sum up* the critical examination, we may say that some of its underlying ideas, but not the independence postulate as a whole is supportable. Its weak point is its universal claim. The Church-Turing thesis, having a less universal character, seems way better justifyable.

4 Forbidden information

In this chapter, we want to discuss Levin's assertions in "Forbidden Information" [Lev10].[1] Particularly, we will proof the forbidden information theorem 4.1 and use it to argue for the forbidden information thesis 4.14, which concerns consistent completions of PA. We already outlined the argumentation in the introduction. Besides, we want to show some consequences, that are not explicitily mentioned in "Forbidden Information".

4.1 The forbidden information theorem

The forbidden information theorem 4.1 should be seen against the background of the independence postulate which we discussed in the previous chapter. The postulate states that a sequence β that is generated by any locatable physical process may not have infinite mutual information with a sequence α that is definable in $\mathbb{N}$. The forbidden information theorem 4.1 will show, that every total extension U of the universal partial computable predicate $\mathbf{u}$ has infinite information with the halting probability Ω (Definition 2.38), which is definable in $\mathbb{N}$. Hence, if we accept the independence postulate and take $\beta = U$ and $\alpha = \Omega$, it is "forbidden" that any locatable physical process generates a total extension of $\mathbf{u}$.

Let us first clarify some notation for the theorem and its proof. Remember that $\mathbf{u}$ denotes the universal partial computable predicate (Definition 2.32).

[1] We will not discuss some parts of that papers, such as "Proposition 1" and the tiling example. They are not relevant for the central argumentation.

In the text of the theorem, requiring $U_n \in \{0,1\}^* \cup \{0,1\}^\infty$ to be a total extension of $\mathbf{u}$ on $\{0,1\}^{n+d}$ means that $U_n(m) = \mathbf{u}(m)$, for all $m \in \{0,1\}^{n+d}$ such that $\mathbf{u}(m)\downarrow$. However, during the proof, we will identify predicates on $\{0,1\}^k$ simply with strings in $\{0,1\}^{2^k}$.

We already mentioned above the halting probability Ω. Keep in mind, that Ω is a Martin-Löf random, left-c.e. real.

4.1.1 The theorem

Theorem 4.1 (Forbidden information theorem[2])**.** *Let ρ be a Martin-Löf random, left-c.e. real. For all $n \in \mathbb{N}$, let $U_n \in \{0,1\}^* \cup \{0,1\}^\infty$ be a total extension of $\mathbf{u}$ on $\{0,1\}^{n+d}$, for some constant d defined in the proof; and $\rho_n := \rho\restriction(n + 2 \cdot \lfloor \log n \rfloor)$. Then*

$$\mathbf{I}(U_n : \rho_n) \overset{+}{>} n - \mathbf{K}(\mathbf{K}(n)|n) - c_\rho,$$

for all $n \in \mathbb{N}$, and for some constant c_ρ defined in the proof.

Remark 4.2. The following points should be mentioned with respect to the above theorem and its proof.

(i) Levin defines ρ_n as $\rho\restriction(n+\mathbf{K}(n))$ but we were only able to prove the statement with a length such that n is computable from ρ_n, and that grows faster than $n + \mathbf{K}(n)$.

(ii) The constant d seems necessary for a proper proof. Levin drops it, probably for reasons of convenience.

(iii) Without proof we remark that

$$\mathbf{K}(\mathbf{K}(n)|n) \overset{+}{<} \log \ell(n) \overset{+}{<} \log\log n,$$

for all $n \in \mathbb{N}$.

[2]Levin calls this theorem simply "Theorem 1" in "Forbidden Information". We entitle it "forbidden information theorem" since it is the main theorem of the paper. And with a proper name, we are able to easier refer to it. Moreover, in combination with the independence postulate, it "forbids" the existence of a total extensions of $\mathbf{u}$ in reality.

4.1.2 The proof

Remark 4.3. The basic steps of the proof are the following. Let $n \in \mathbb{N}$.

(Step 1) We show that $\mathbf{I}$ has the lower bound

$$\mathbf{I}(U_n : \rho_n) \stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(Q_n|U_n) - \mathbf{K}(Q_n|\rho_n),$$

for any string Q_n. So it suffices to find a single Q_n, such that the latter expression gets big enough.

(Step 2) Q_n being a total extension of a partial computable predicate P on $\{0,1\}^n$ via U_n is a good candidate: On the one hand, for such a Q_n, we have $\mathbf{K}(Q_n|U_n) \stackrel{+}{<} \mathbf{K}(n)$.

(Step 3) And on the other hand, we can construct P in a way, that knowing ρ_n lets us compress Q_n to size $\mathbf{K}(Q_n|\rho_n) \stackrel{+}{<} \mathbf{K}(Q_n|n) - n$ (using the fact that an approximation of ρ_n necessarily takes longer than some specific approximation of P).

(Step 4) Plugging the estimations of Step 2 and 3 into the inequality in Step 1, we obtain the claimed inequality.

Step 1

Lemma 4.4. *We have*

$$\mathbf{K}(Q) \stackrel{+}{>} \mathbf{K}(Q|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n), \tag{4.1}$$

for all $n \in \mathbb{N}$, *and* $Q \in \{0,1\}^{2^n}$.

Proof. Due to Theorem 3.3 we have for all $n \in \mathbb{N}$, and $Q \in \{0,1\}^{2^n}$

$$\mathbf{K}(Q|n^*) + \mathbf{K}(n) \stackrel{+}{=} \mathbf{K}(Q,n) \stackrel{+}{=} \mathbf{K}(Q).$$

Adding $\mathbf{K}(n^*|n) \stackrel{+}{=} \mathbf{K}(\mathbf{K}(n)|n)$, we obtain

$$\mathbf{K}(Q|n^*) + \mathbf{K}(n^*|n) + \mathbf{K}(n) \stackrel{+}{=} \mathbf{K}(Q) + \mathbf{K}(\mathbf{K}(n)|n).$$

We have $\mathbf{K}(Q|n) \stackrel{+}{<} \mathbf{K}(Q|n^*) + \mathbf{K}(n^*|n)$, due to Theorem 2.11, hence

$$\mathbf{K}(Q|n) + \mathbf{K}(n) \stackrel{+}{<} \mathbf{K}(Q) + \mathbf{K}(\mathbf{K}(n)|n).$$

□

We have for all $n \in \mathbb{N}$

$$\begin{aligned}
\mathbf{I}(U_n : \rho_n) &\stackrel{+}{=} \log \sum_{x,y \in \mathbb{N}} \mathbf{m}(x|U_n) \cdot \mathbf{m}(y|\rho_n) \cdot 2^{\mathbf{I}(x:y)} \\
&\stackrel{+}{>} \mathbf{K}(Q_n) - \mathbf{K}(Q_n|U_n) - \mathbf{K}(Q_n|\rho_n) \\
&\stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(Q_n|U_n) - \mathbf{K}(Q_n|\rho_n),
\end{aligned}$$

by taking Q_n for x and y and due to the above lemma.

Step 2

We define a partial computable predicate $P\colon \{0,1\}^* \to \{0,1\}$. For this purpose we construct the following Turing machine M, and set $P := \varphi_M$.

For $n \in \mathbb{N}$, we define M inductively for inputs $0^n, 0^{n-1}1, \ldots, 1^n$:

$\underline{x = 0^n}$: M simultaneously enumerates lower bounds of

$$m_{0^n,i} := \sum_{Q \in \{0,1\}^{2^n}} \mathbf{m}(Q|n),$$

for $i = 0, 1$. Let $i_0 \in \{0,1\}$ such that the lower bound of m_{0^n,i_0} first exceeds 2^{-n}. M outputs $1 - i_0$.

$\underline{x-1 \curvearrowright x}$: M first computes $M(0^n), M(0^{n-1}1), \ldots, M(x-1)$. If all computations converge, let

$$\widehat{Q}_{x,i} := \big\{Q \in \{0,1\}^{2^n} : \ Q(y) = M(y) \text{ for all } y \in \{0^n, \ldots, x-1\}, \text{ and } Q(x) = i\big\}.$$

M simultaneously enumerates lower bounds of

$$m_{x,i} := \sum_{Q \in \widehat{Q}_{x,i}} \mathbf{m}(Q|n),$$

for $i = 0, 1$. Again, let $i_0 \in \{0,1\}$ such that the lower bound of m_{x,i_0} first exceeds 2^{-n}. M outputs $1 - i_0$. ⌟

Similar as in the construction of M, but now with respect to the final M that is defined for all inputs $x \in \{0,1\}^*$ (note that defined here does not mean that M converges), let

$$\widehat{Q}_{x,i} := \big\{Q \in \{0,1\}^{2^{\ell(x)}} : Q(y) = M(y) \text{ for all } y \in \{0^{\ell(x)}, \ldots, x-1\} \text{ such that } M(y)\downarrow, \text{ and } Q(x) = i\big\},$$

and

$$m_{x,i} := \sum_{Q \in \widehat{Q}_{x,i}} \mathbf{m}(Q|n),$$

for all $x \in \{0,1\}^*$ and $i \in \{0,1\}$.

We can make the following observations. Let $n \in \mathbb{N}$.

(i) If $M(x)$ diverges for some $x \in \{0,1\}^n$, then $M(y)$ diverges for all $y \in \{0,1\}^n$, with $y > x$, too.

(ii) For all $x \in \{0,1\}^n$ such that $M(x+1)$ converges, we have

$$m_{x+1,0} + m_{x+1,1} < m_{x,0} + m_{x,1} - 2^{-n},$$

since by construction, $M(x)$ "diagonalizes" against Q's of total measure at

least 2^{-n}. Moreover, since we have

$$m_{0^n,0} + m_{0^n,1} = \sum_{Q \in \{0,1\}^n} \mathbf{m}(Q|n) < 1,$$

there must be some least $x \in \{0,1\}^n$, so that $m_{x,i} < 2^{-n}$ for both, $i = 0$ and $i = 1$. We denote this least x by x_n for the rest of the proof. So M diverges on $x_n, \ldots, 1^n$.

(iii) Particularly,

$$m_{x_n,0} + m_{x_n,1} < 2 \cdot 2^{-n}. \tag{4.2}$$

Now we have constructed the partial computable predicate P. Since $\mathbf{u}$ is the universal partial computable predicate, there is some string $p \in \{0,1\}^*$, so that $\mathbf{u}(p^\frown x) \cong P(x)$ for all $x \in \{0,1\}^*$. Let

$$d := \ell(p).$$

Then given n

$$Q_n(x) := U_n(p^\frown x), \text{ for all } x \in \{0,1\}^n,$$

extends P to a total predicate on $\{0,1\}^n$, by assumption. As we can compute Q_n from U_n knowing n we have

$$\mathbf{K}(Q_n|U_n) \stackrel{+}{<} \mathbf{K}(n). \tag{4.3}$$

Step 3

For $n \in \mathbb{N}$ we define

$$\widehat{Q}_{x_n} := \big\{Q \in \{0,1\}^{2^n} : \ Q(y) = P(y) \text{ for all } y \in \{0^n, \ldots, x_n - 1\}\big\}.$$

So $\widehat{Q}_{x_n} = \widehat{Q}_{x_n,0} \cup \widehat{Q}_{x_n,1}$ and $Q_n \in \widehat{Q}_{x_n}$.

Lemma 4.5. *We have*

$$\mathbf{K}(Q|x_n) \ \stackrel{+}{<} \ \mathbf{K}(Q|n) - n,$$

for all $n \in \mathbb{N}$, *and* $Q \in \widehat{Q}_{x_n}$.

Proof. Let $n \in \mathbb{N}$ be arbitrary. Note that from x_n we can both, compute $\widehat{Q}_{x_n} \subset \{0,1\}^{2^n}$ and upper semicompute $w \mapsto \mathbf{K}(w|n)$ (since $n = \ell(x_n)$). We use these facts to build a KC set to prove the above inequality.

The machine M_1 with $\overline{x_n}$ on its oracle tape works in the following way. $M_1^{\overline{x_n}}$ internally enumerates simultaneously the sets $\{k : k \geq \mathbf{K}(Q|n)\}$, for all $Q \in \widehat{Q}_{x_n}$. Whenever for some Q, a *new* upper bound d was enumerated for $\mathbf{K}(Q|n)$, then M outputs a new KC request $\langle d - n + 2, Q\rangle$.

Let $(\langle d_i - n + 2, \widetilde{Q}_i\rangle)_{i\in\mathbb{N}}$ denote the sequence computed by $M_1^{\overline{x_n}}$. We have

$$\begin{aligned}
\sum_{i\in\mathbb{N}} 2^{-d_i} &= \sum_{Q\in\widehat{Q}_{x_n}} \sum_{i:\,\widetilde{Q}_i=Q} 2^{-d_i} \\
&\leq \sum_{Q\in\widehat{Q}_{x_n}} \sum_{j\in\mathbb{N}} 2^{-(\mathbf{K}(Q|n)+j)} \\
&\leq 2 \sum_{Q\in\widehat{Q}_{x_n}} 2^{-\mathbf{K}(Q|n)} \\
&< 2^{-n+2},
\end{aligned}$$

due to inequality 4.2. So $(\langle d_i - n + 2, \widetilde{Q}_i\rangle)_{i\in\mathbb{N}}$ is in fact a KC set and we can apply the KC theorem 2.14. We get a prefix-free oracle machine M_2 such that by Theorem 2.9

$$\mathbf{K}(Q|x_n) \stackrel{+}{<} \mathbf{K}_{M_2}(Q|x_n) = \mathbf{K}(Q|n) - n + 2,$$

for all $Q \in \widehat{Q}_{x_n}$ (with the additive constant implicit in "$\stackrel{+}{<}$" depending on M_2).

□

Lemma 4.6. *There is a Turing machine, depending on* ρ, *that computes* x_n *on input* ρ_n, *for all* $n \in \mathbb{N}$. *Particularly*

$$\mathbf{K}(Q_n|\rho_n) \leq \mathbf{K}(Q_n|x_n) + c_\rho,$$

for all $n \in \mathbb{N}$, *and some constant* c_ρ *depending on* ρ.

Proof. Let $(q_k)_{k\in\mathbb{N}}$ be a monotonically increasing, computable sequence, such that $\lim_{k\to\infty} q_k = \rho$. By the construction of M in Step 1, we can consider x_n as a (uniformly in n) left-c.e. real; for this purpose we identify $x \in \{0,1\}^n$ with the real $0.x$ for the moment. So for each n there is a monotonically increasing sequence $(r_{n,k})_{k\in\mathbb{N}}$, such that $\lim_{k\to\infty} r_{n,k} = x_n$; and $\langle n,k\rangle \mapsto r_{n,k}$ is computable.

Hence, there is an $n_0 \in \mathbb{N}$, such that for all $n \geq n_0$, we have: $r_{n,k} \geq x_n$ not after $q_k \geq \rho_n$, for increasing k. For a proof by contradiction assume there was an infinite $N \subset \mathbb{N}$, such that for all $n \in N$, $q_k \geq \rho_n$ before $r_{n,k} \geq x_n$, for increasing k.

We construct the following oracle machine M_1. On input x_n^*, M_1 enumerates the sequence $(r_{n,k})_{k\in\mathbb{N}}$ till the first index k_0, such that $r_{n,k_0} \geq x_n$. Let s be the biggest rational in $\{0,1\}^{n+2\cdot\lfloor\log n\rfloor}$, such that $s \leq q_{k_0}$. M_1 outputs s.

For all $n \in N$, M_1 computes ρ_n on input x_n^*.[3] Therefore

$$\mathbf{K}(\rho_n) \stackrel{+}{<} \mathbf{K}_{M_1}(\rho_n) + c_{M_1} \leq \ell(x_n^*) + c_{M_1} \stackrel{+}{<} n + \mathbf{K}(n) + c_{M_1},$$

for the infinitely many $n \in N$ (due to Theorem 2.13) and some constant c_{M_1}. This yields a contradiction to the fact that $\rho_n = \rho\restriction(n + 2\cdot\lfloor\log n\rfloor)$ and ρ is Martin-Löf random (due to Theorem 2.13, we have $\mathbf{K}(n) \stackrel{+}{<} \log n + 2\log\log n$, for all $n \in \mathbb{N}$).

So we have for all $n \geq n_0$: $r_{n,k} \geq x_n$ not after $q_k \geq \rho_n$, for increasing k. Due to this fact, we can build a prefix-free machine M_2, similar to M_1, that for all $n \geq n_0$ computes x_n from input ρ_n. Note that M_2 depends on ρ. □

[3] Because (for $n \in N$) on the one hand, if $s < \rho_n \leq q_{k_0}$ this contradicts the construction, since $\rho_n \in \{0,1\}^{n+2\cdot\lfloor\log n\rfloor}$. And on the other hand, if $s > \rho_n$, then $\rho_n + 0.0^{(n+2\cdot\lfloor\log n\rfloor-1)}1 \leq q_{k_0} \leq \rho$, which is a contradiction to ρ being irrational and thus having a unique binary representation.

Step 4

We obtain for all $n \in \mathbb{N}$

$$\begin{aligned}
\mathbf{I}(U_n : \rho_n) &\stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(Q_n|U_n) - \mathbf{K}(Q_n|\rho_n) \\
&\stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(n) - \mathbf{K}(Q_n|\rho_n) \\
&\stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(n) - [\mathbf{K}(Q_n|n) - n - c_\rho] \\
&= n - \mathbf{K}(\mathbf{K}(n)|n) - c_\rho,
\end{aligned}$$

where the first inequality is due to Step 1, the second one due to Step 2 (inequality 4.3), and the third one due to Step 3 (Lemma 4.6 and 4.5).

□

4.2 Some implications

We proceed with discussing a version of the forbidden information theorem 4.1 for two infinite sequences, and two probabilistic assertions that follow from the forbidden information theorem 4.1.

4.2.1 A version for two infinite sequences

The following corollary will be important for us when we come back to complete, consistent extensions of PA in the last part of this chapter, since it makes the independence postulate directly applicable. Levin does not mention this corollary

Corollary 4.7. *Let ρ be a Martin-Löf random, left-c.e. real. Let U be a total extension of* $\mathbf{u}$*. Then we have*

$$\hat{\mathbf{I}}(U : \rho) = \infty.$$

Proof. Have in mind the proof of the forbidden information theorem 4.1 and the notations we used there. Particularly, we proved Lemma 4.6 stating that $\mathbf{K}(Q_n|\rho_n) \leq \mathbf{K}(Q_n|x_n) + c_\rho$, for all $n \in \mathbb{N}$.

Obviously, we have $\mathbf{K}(Q_n|\langle \rho, \overline{n}0\ldots\rangle) \stackrel{+}{<} \mathbf{K}(Q_n|\rho_n)$. (We have to write "$\langle \rho, \overline{n}0\ldots\rangle$" instead of "$\langle \rho, n\rangle$", since we did not define the pairing function for a mix of finite and infinite sequences.) Furthermore, there is a prefix-free oracle machine that, with ρ on its oracle tape and input $n^*{}^\frown(Q_n)^{*(\langle \rho, \overline{n}0\ldots\rangle)}$, computes Q_n, so

$$\mathbf{K}(Q_n|\rho) \stackrel{+}{<} \mathbf{K}(n) + \mathbf{K}(Q_n|\langle \rho, \overline{n}0\ldots\rangle).$$

Together, we obtain

$$\mathbf{K}(Q_n|\rho) \stackrel{+}{<} \mathbf{K}(n) + \mathbf{K}(Q_n|x_n) + c_\rho.$$

Therefore, we get for all $n \in \mathbb{N}$

$$\begin{aligned}
\hat{\mathbf{I}}(U:\rho) &\stackrel{+}{=} \log \sum_{x,y\in\mathbb{N}} \mathbf{m}(x|U)\cdot\mathbf{m}(y|\rho)\cdot 2^{\mathbf{I}(x:y)} \\
&\stackrel{+}{>} \mathbf{K}(Q_n) - \mathbf{K}(Q_n|U) - \mathbf{K}(Q_n|\rho) \\
&\stackrel{+}{>} \mathbf{K}(Q_n) - \mathbf{K}(Q_n|U) - \mathbf{K}(Q_n|x_n) - \mathbf{K}(n) - c_\rho \\
&\stackrel{+}{>} [\mathbf{K}(Q_n|n) + \mathbf{K}(n) - \mathbf{K}(\mathbf{K}(n)|n)] - \mathbf{K}(n) - [\mathbf{K}(Q_n|n) - n] - \mathbf{K}(n) - c_\rho \\
&\stackrel{+}{>} n - \mathbf{K}(\mathbf{K}(n)|n) - \mathbf{K}(n) - c_\rho \\
&\stackrel{+}{>} n - \log\log n - 2\log n - c_\rho,
\end{aligned}$$

where the third last inequality is due to Lemma 4.4, inequality 4.2 (U meets the requirements set up for U_n) and Lemma 4.5; and the last inequality is due to Remark 4.2 and Theorem 2.13.

So $\hat{\mathbf{I}}(U:\rho) = \infty$. □

4.2.2 Two probabilistic assertions

We proceed with discussing two probabilistic assertion, one with respect to partial, and one with respect to total extensions of $\mathbf{u}$. They are particularly interesting when interpreting them with respect to randomized algorithms.

The following theorem is not stated in "Forbidden Information". However, it is similar to "Corollary 1" but stronger, and indicated in footnote 7 in the paper. Keep in mind the definitions of d and c_ρ in the proof of the forbidden information theorem 4.1.

Theorem 4.8 ("Corollary 1")**.** *Let $(P_n)_{n\in\mathbb{N}}$ be a family of uniformly lower semi-computable discrete measures. Then, given $n \in \mathbb{N}$, for any $U_n \in \{0,1\}^{2^{n+d}}$ that is a total extension of $\mathbf{u}$ on $\{0,1\}^{n+d}$, we have*

$$P_n(U_n) \overset{*}{<} 2^{-n},$$

with the multiplicative constant implicit in "$\overset{*}{<}$" depending on P.

By Section 2.1.4, for a randomized algorithm, the probability that it outputs y on input n is computable. So we can apply the above theorem and obtain the following corollary.

Corollary 4.9. *The probability for a randomized algorithm to compute a total extension of $\mathbf{u}$ on $\{0,1\}^{n+d}$, on input n, is less than or equal to 2^{-n}, up to a multiplicative constant depending on the algorithm.*

Proof of Theorem 4.8. For the following argumentation, have in mind the proof of the forbidden information theorem 4.1 and the notations we used there.

Since $\langle x, n\rangle \mapsto P_n(x)$ is computable, we have $P_n(x) \overset{*}{<} \mathbf{m}(x|n)$, for all $n \in \mathbb{N}$, $x \in \{0,1\}^*$, with the multiplicative constant implicit in "$\overset{*}{<}$" depending on P (see Remark 2.20). Hence

$$P_n(x) \cdot 2^{\mathbf{I}(x:y|n)} \overset{*}{<} 2^{-\mathbf{K}(x|n)+\mathbf{K}(x|n)-\mathbf{K}(x|y,\mathbf{K}(y|n),n)} \leq 1, \tag{4.4}$$

for all $n \in \mathbb{N}$, $x, y \in \{0,1\}^*$, due to Lemma 3.5.

The forbidden information theorem 4.1 works with U_n the way we defined it now, too. As in the proof of that theorem, let Q_n be the total extension of the partial computable predicate P on $\{0,1\}^n$ via U_n. So particularly, Q_n is uniformly computable from U_n and vice versa. Therefore $\mathbf{I}(x_n : U_n|n) \stackrel{+}{=} \mathbf{I}(x_n : Q_n|n)$, for all $n \in \mathbb{N}$ (Theorem 3.10 can be easily modified to hold for the conditional version of $\mathbf{I}$ as well).

Furthermore, we have

$$\begin{aligned}\mathbf{I}(x_n : Q_n|n) &= \mathbf{K}(x_n|n) + \mathbf{K}(Q_n|n) - \mathbf{K}(x_n, Q_n|n) \\ &\stackrel{+}{>} \mathbf{K}(x_n|n) + \mathbf{K}(Q_n|x_n) + n - \mathbf{K}(x_n, Q_n|n) \\ &\stackrel{+}{>} \mathbf{K}(x_n, Q_n|n) + n - \mathbf{K}(x_n, Q_n|n) \\ &= n,\end{aligned}$$

for all $n \in \mathbb{N}$, where the first inequality is due to Lemma 4.5 and the second one due to Theorem 2.11.

Together with inequality 4.4, we obtain

$$1 \stackrel{*}{>} P_n(U_n) \cdot 2^{\mathbf{I}(U_n : x_n|n)} \stackrel{*}{>} P_n(U_n) \cdot 2^n,$$

for all $n \in \mathbb{N}$. □

We turn to a probabilistic assertion regarding total extensions of $\mathbf{u}$. It is important to mention, that the below Theorem 4.10 was already proved by Jokusch and Soare [JS72] with respect to the uniform measure (Lebesgue measure). It may be that their proof can be easily extended to arbitrary computable probability measures and it seems likely that a proof for arbitrary computable probability measures was already published somewhere, though we are not aware of such work. Anyway, we state Theorem 4.10, since it can be very easily derived from the forbidden information theorem 4.1 in combination with the second independence

conservation inequality for infinite sequences (Theorem 3.21). Note that Levin does not mention Theorem 4.10 in "Forbidden Information".

Let

$$E := \{\alpha \in \{0,1\}^\infty : \text{ there is a total extension of } \mathbf{u} \text{ that is computable in } \alpha\}.$$

Theorem 4.10. *Let P be a computable continuous probability measure. Then $P(E) = 0$.*

Proof. By Corollary 4.7 and Theorem 3.21, we have $\hat{\mathbf{I}}(\alpha : \Omega) = \infty$, for all $\alpha \in E$. Therefore, by Corollary 3.23, we obtain $P(E) = 0$. □

Remember the discussion of randomized operators in Section 2.1.4. Now we consider a randomized operator that gets no input on its oracle tape and tries to guess a total extension of $\mathbf{u}$.

Corollary 4.11. *The probability that a randomized operator with no input computes a total extension of $\mathbf{u}$ is zero.*

Proof. Let (M, Q) be a randomized operator that gets no input. As Q is a computable distribution, we have $Q(E) = 0$. But

$$E \supset \{\alpha \in \{0,1\}^\infty : M \text{ computes a total extension of } \mathbf{u} \text{ with } \alpha \text{ on its randomness tape}\}.$$

□

4.3 Consequences for the completion of PA

Let us come back to the problem of finding a complete, consistent extension of PA, that we discussed in the introduction and in Section 2.4.2. Gödel showed with his incompleteness theorem that the completion is impossible if we only allow

computable methods. However, what the incompleteness theorem does not rule out is the possibility of such a completion by other realistic means, such as random or other processes (particularly when we keep in mind that there are continuum many consistent completions of PA[4]).

Let us give an example. We consider a problem that is - at the first sight - similar to the completion task, namely generating Martin-Löf random sequences. These sequences are not computable.[5] However, by Definition 2.23 and Theorem 2.24, the set of Martin-Löf random reals has Lebesgue measure one. This means that an independent, uniformly distributed sequence of $\{0,1\}$-valued random variables is Martin-Löf random with probability one (as the Lebesgue measure is the product measure on $\{0,1\}^\infty$ that is induced by the uniform distribution on $\{0,1\}$). So theoretically, we can use a randomized method, namely flipping a fair coin, to generate a solution for a problem that has no computable solutions.

One may wonder, if randomized or other realistic methods could generate a consistent completion of PA, too. In this section we want to argue, that this is not the case, i.e. we can extend Gödel's assertion to noneffective methods.

Note that, in what follows, it will be necessary to distinguish between *"Gödel's incompleteness theorem"* which refers to the formal assertion proved by Gödel, on the one hand, and what we want to call *"Gödel's thesis"*, namely that there is no anyhow effectively calculable consistent completion of PA, on the other hand.

Now first, we want to mention an extension of Gödel's incompleteness theorem to randomized operators that can be easily derived from the forbidden information theorem 4.1. And second and most important, we will present the forbidden information thesis 4.14, the extension of Gödel's thesis we already mentioned in the introduction.

[4]Roughly speaking, this can be seen as follows. For each consistent extension of PA that contains only a finite number of additional axioms, we have the binary choice whether we take the "next" undecidable sentence (that exists due to the incompleteness theorem) or its contrary as new axiom. So we have a countable number of binary choices to determine a complete, consistence extension of PA and therefore continuum many such extensions (since all these extensions differ from each other).

[5]Otherwise the Kolmogorov complexity of an initial segment of length n would be of order $\mathbf{K}(n)$, which contradicts the definition.

4.3.1 Extending Gödel's incompleteness theorem to randomized operators

As mentioned in the example above, there are random processes that generate incomputable sequences, namely Martin-Löf random ones, with probability one. Fitting well into the context, we state the following theorem and its corollary which show that we do not have such random processes with respect to consistent completions of PA, at least if we only allow processes that are induced by randomized operators, i.e. that have computable distributions. The statements can be proved similar to Theorem 4.10 and Corollary 4.11, using Theorem 2.34.

Theorem 4.12. *Let P be a computable continuous probability measure. Then*

$$P(\{\alpha \in \{0,1\}^\infty : \ \alpha \textit{ is } \text{PA-}\textit{complete}\}) = 0.$$

Corollary 4.13. *The probability that a randomized operator with no input computes a complete, consistent extension of* PA *is zero.*

It is important to mention that, as was the case with Theorem 4.10, Theorem 4.12 was already proved by Jokusch and Soare [JS72] with respect to the uniform measure (Lebesgue measure). It seems very likely that a generalization to arbitrary computable probability measures has already been proved too, though we could not find any publication containing such a generalization. We stated the theorem anyway since it suits into the context, and since its proof - for arbitrary computable probability measures - is so immediate by the forbidden information theorem 4.1 in combination with the second independence conservation inequality for infinite sequences (Theorem 3.21).

4.3.2 Extending Gödel's thesis using the independence postulate

We go one step further and come to the central thesis we review in the present work. The Church-Turing thesis gives Gödel's incompleteness theorem a meaning

beyond the scope of the formal language of mathematics: there is no anyhow effectively calculable consistent completion of PA. As already mentioned, we refer to this assertion as Gödel's thesis. Similarly, we can combine the forbidden information theorem 4.1 with the independence postulate to obtain a fundamental thesis regarding the physical world that is an extension of Gödel's thesis.

Thesis 4.14 (Forbidden information thesis[6])**.** No sequence that is generated by any locatable physical process is a consistent completion of PA.

To be more precise, the argumentation for this thesis is that each consistent completion α of PA computes a total extension of $\mathbf{u}$ (by Theorem 2.34) and therefore has infinite mutual information with the halting probability Ω, i.e. $\hat{\mathbf{I}}(\alpha : \Omega) = \infty$ (due to Corollary 4.7 and the independence conservation inequality for algorithmic operators, Theorem 3.21). Thus, if we accept the independence postulate, α can not be a sequence generated by any locatable physical process.

Keep in mind, that the weaknesses of the inependence postulate, which we discussed in Section 3.3.3, completely transfer to the forbidden information thesis 4.14!

Levin sees the forbidden information thesis 4.14 particularly as a reply to the following statement by Gödel:

> "Namely, it turns out that in the systematic establishment of the axioms of mathematics, new axioms, which do not follow by formal logic from those previously established, again and again become evident. It is not at all excluded by the negative results mentioned earlier that nevertheless every clearly posed mathematical yes-or-no question is solvable in this way. For it is just this becoming evident of more and more

[6] We call this thesis "forbidden information thesis" since it is the central thesis of the paper "Forbidden Information" (although Levin does not explicitly mention it in this precise form); and moreover, it "forbids" the consistent completion of PA in reality. Levin did not give any title to this thesis.

> new axioms on the basis of the meaning of the primitive notions that a machine cannot imitate."[7]

In Levin's opinion, the forbidden information thesis 4.14 "dous[es] Gödel's hope" ([Lev10], p. 7) concerning the realizability of a consistent completion of PA by non-algorithmic means.

[7] Kurt Gödel. The modern development of the foundations of mathematics in the light of philosophy. In: Kurt Gödel. Collected Works. Volume III. Oxford University Press, 1961. Cited after [Lev10], p. 1.

5 Conclusion

Our objective was to completely and critically elaborate Levin's argumentation for an extension of Gödel's incompleteness assertion in "Forbidden Information" [Lev10]. For the most part, we achieved this objective. We were able to present proofs for all mathematical arguments, except for the second conservation inequality for infinite sequences (and regarding the forbidden information theorem, we had to slightly vary its content to be able to proof it). This was the main part of the work. And we critically discussed the non-mathematical arguments by weighing thoroughly their pros and cons.

So we can confirm the validity of all mathematical arguments (with the possible exception of the second conservation inequality for infinite sequences). However, the non-mathematical independence postulate could not withstand a critical examination as a whole, whereby some of its underlying ideas seem justifiable. The weaknesses of the inependence postulate completely transfer to the forbidden information thesis.

We already mentioned in the introduction, that one purpose we pursued with the present work is to promote philosophical knowledge. Kant wrote in his "Critique of Pure Reason" ([Kan29], A51):

> "Thoughts without content are empty, intuitions without concepts are blind. [...] Only through their union [i.e., the union of concepts and intuitions] can knowledge arise."

Within modern mathematics, we can observe a remarkable elaboration of mathematical concepts in a strictly formal way, making intuitions "less blind ". However at the same time, most of these concepts do *only* subsist in the formal language

and therefore are pretty far away from intuition.[1] Hence, many results regarding those concepts are rather "empty" in the above philosophical sense. More precisely, their potential for *direct* philosophical insight is little.[2]

Against this background, we regard the forbidden information thesis and its underlying argumentation as a good example of concepts and assertions that are formal and quite abstract - that can however *directly* be brought together with *substantial* real-world interpretations. As already mentioned in the introduction, the philosophical content of the forbidden information thesis is the attempt of establishing a definite limit for formal mathematical knowledge.

[1] In contrast to everyday language and human thoughts, the formal language needs no interpretations, no contents. This can particularly be seen by the fact that machines can solve major mathematical problems, such as finding proofs for conjectures.

[2] Of course, there are many very abstract formal mathematical results, that for example contribute to physical and thereby also to philosophical knowledge. However, in many cases they do not have any straightforward interpretations themselves, only in conncetion with a complex theory.

Bibliography

[DH10] Rodney Downey and Denis Hirschfeldt. *Algorithmic Randomness and Complexity*. Springer, 2010.

[EFT94] Heinz-Dieter Ebbinghaus, Jörg Flum, and Wolfgang Thomas. *Mathematical Logic*. Springer, second edition, 1994.

[Gac] Peter Gacs. Lecture notes on descriptional complexity and randomness. Lecture notes. Available online at `http://www.cs.bu.edu/faculty/gacs/papers/ait-notes.pdf` (accessed 13 September 2012).

[Gan80] Robin Gandy. Church's Thesis and Principles for Mechanisms. *Studies in Logic and the Foundations of Mathematics*, 101:123–148, 1980.

[HW12] Denis Hirschfeldt and Rebecca Weber. Finite Self-Information. *Computability*, 1:85–98, 2012.

[JS72] Carl Jockusch and Robert Soare. Pi01 classes and degrees of theories. *Transactions of the American Mathematical Society*, 172:33–56, 1972.

[Kan29] Immanuel Kant. *Critique of Pure Reason*. MacMillan, 1929. Translated by Norman Kemp Smith.

[Kol68] Andrey Kolmogorov. Three approaches to the quantitive definition of information. *International Journal of Computer Mathematics*, 2:157–168, 1968.

[Lev74] Leonid Levin. Laws of Information Conservation (Nongrowth) and Aspects of the Foundation of Probability Theory. *Problems of Information Transmission*, 10(3):206–210, 1974.

[Lev80] Leonid Levin. *A concept of independence with applications in various fields of mathematics*. MIT, Laboratory for Computer Science, 1980.

[Lev84] Leonid Levin. Randomness Conservation Inequalities; Information and Independence in Mathematical Theories. *Information and Control*, 61(1):15–37, 1984.

[Lev10] Leonid Levin. Forbidden Information. 2010. Preprint (arXiv:cs/0203029v16 [cs.CC]).

[LV08] Ming Li and Paul Vitanyi. *An Introduction to Kolmogorov Complexity and Its Applications*. Springer, 2008.

[Mac03] David MacKay. *Information Theory, Inference, and Learning Algorithms*. Cambridge University Press, 2003.

[Odi92] Piergiorgio Odifreddi. *Classical recursion Theory. The Theory of Functions and Sets of Natural Numbers*. Elsevier, 1992.

[Ste06] Frank Stephan. Martin-Löf Random and PA-complete Sets. *ASL Lecture Notes in Logic*, 27:342–348, 2006.

[Zac09] Richard Zach. Hilbert's Program. *The Stanford Encyclopedia of Philosophy (Spring 2009 Edition)*, 2009. Online encyclopedia entry. Available online at `http://plato.stanford.edu/archives/spr2009/entries/hilbert-program/` (accessed 13 September 2012).

[ZL70] Alexandre Zvonkin and Leonid Levin. The complexity of finite objects and the developement of the concepts of information and randomness by means of the theory of algorithms. *Russian Mathematical Surveys*, 25(6):83–124, 1970.

Erklärung

Hiermit versichere ich, dass ich meine Arbeit selbstständig unter Anleitung verfasst habe, dass ich keine anderen als die angegebenen Quellen und Hilfsmittel benutzt habe, und dass ich alle Stellen, die dem Wortlaut oder dem Sinne nach anderen Werken entlehnt sind, durch die Angabe der Quellen als Entlehnungen kenntlich gemacht habe.

Datum Unterschrift