Computability of the Zero-Error Capacity with Kolmogorov Oracle

Holger Boche∗†
∗Lehrstuhl für Theoretische Informationstechnik
Technische Universität München
†Munich Center for Quantum Science and Technology
Munich, Germany
boche@tum.de

Christian Deppe
Lehr- und Forschungsseinheit für Nachrichtentechnik
Technische Universität München
Munich, Germany
christian.deppe@tum.de

Abstract—The zero-error capacity of a discrete classical channel was first defined by Shannon as the least upper bound of rates for which one transmits information with zero probability of error. The problem of finding the zero-error capacity \( C_0 \), which assigns a capacity to each channel as a function \( \Theta \), which assigns a value to each simple graph. This paper studies the computability of the zero-error capacity. For the computability, the concept of a Turing machine and a Kolmogorov oracle is used. It is unknown if the zero-error capacity is computable in general. We show that in general the zero-error capacity is semi-computable with the help of a Kolmogorov Oracle. Furthermore, we show that \( C_0 \) and \( \Theta \) are computable functions if and only if there is a computable sequence of computable functions of upper bounds, i.e. the converse exist in the sense of information theory, which point-wise converges to \( C_0 \) or \( \Theta \). Finally, we examine Zuiddam’s characterization of \( C_0 \) and \( \Theta \) in terms of algorithmic computability.

Index Terms—zero-error capacity, Kolmogorov Oracle, Turing computability

I. INTRODUCTION

The zero-error capacity of a discrete classical channel is defined as the least upper bound of rates for which one transmits information with zero probability of error. Investigation of the zero-error capacity of discrete memoryless channels (DMCs) has a long tradition in information theory. Shannon already reduced the problem of determining the zero-error capacity of DMCs to a graph theoretical problem. However, it is generally unclear how Shannon’s characterization can be used to compute the zero-error capacity. The zero-error capacity is not defined except for special cases even for simple graphs. In this work we investigate the algorithmic predictability of the zero-error capacity. In general, it is not even known if \( \Theta(G) \) is a computable number for every graph \( G \), which is a strictly weaker assertion than the existence of an algorithm that computes \( \Theta(G) \) in recursive dependence of \( G \). We want to consider the Turing computability of the zero-error capacity. The concept of a Turing machine provides fundamental performance limits for today’s digital computers. Turing machines have no limitations on computational complexity, have unlimited computing capacity and storage, and execute programs completely error-free. They provide fundamental performance limits for digital computers and they are the ideal concept to decide whether or not a function (here the zero-error capacity) is effectively computable. Surprisingly, in information theory the question of Turing computability has attracted little attention in the past. For the concept of the Turing machine see [Tur36], [Tur37], [Wei00]. As described above, Shannon reduced the problem of determining the zero-error capacity of DMCs to a graph theoretical problem. It turns out, that one has to find the maximum independent set of a family of graphs to compute the zero-error capacity. Most important results concerning zero-error information theory can be found in the survey paper [KO98]. Since finding the maximum independent set is a difficult problem, it is unlikely that we will find an efficient algorithm for finding the independence number. We will show that the zero-error capacity is semi-computable if we allow the Kolmogorov oracle. In Section II we introduce an enumeration of simple graphs and give further basic definitions and notations of graph theory and computability theory. In Section III we introduce a Kolmogorov oracle and show that with this oracle we can compute the zero-error capacity up to any given accuracy. We characterize the Shannon Zero-Error Capacity in Section IV and state some results of [Zui19], discuss the Strassen preorder following [Str88] and introduce the asymptotic spectrum of graphs. Finally, we show the semi-decidability with an oracle of a binary relation on graphs.

II. BASIC DEFINITIONS AND CONCEPTS

We will define an enumeration of the simple graphs. We will use the adjacency matrix. First consider all simple graphs with \( n \) vertices. The graph can be described by the values of the adjacency matrix \( A \) by knowing \( a_{ij} \) with \( i < j \in [n] \), because \( A \) is symmetric and the diagonal only has the entries 0. Therefore, we can represent all elements of \( G(n) \) by the binary vector

\[
a_{\frac{n^2 - n}{2}} = (a_{12}, a_{13}, \ldots, a_{1n}, a_{23}, a_{24}, \ldots, a_{2n}, \ldots, a_{n-1n}) \in \{0, 1\}^{\frac{n^2 - n}{2}}
\]

Holger Boche and Christian Deppe were supported by the Bundesministerium für Bildung und Forschung (BMBF) through Grants 16KR10118K and 16KR10117K. H. Boche was also supported in part by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) within Germany’s Excellence Strategy EXC-2111 390814868 and within the Gottfried Wilhelm Leibniz program under Grant BO 1734/20-1.
The zero-error capacity of $C_0(W)$ and $\Theta(G)$ are computable real numbers at all. There are, of course, computable, monotonically increasing sequences of rational numbers, each which converge to a finite limit value, but for which the limit values are not computable numbers. Therefore, the convergence is not effective (see [Spe49]).

We denote with $C_k \in G$ the graph with the vertex set $V_k = \{0, 1, \ldots, k-1\}$ and the edge set $E_k = \{\left\{u, u \oplus k_1\right\}\}$, where $\oplus_k$ denotes the addition modulo $k$. $C_k$ denotes the set of all isomorphic graphs to $C_k$. We now list some properties that are important for our later considerations.

1. For the maximum independent set in a graph, it holds (by definition of $\otimes$) that $\alpha(G^{2n}) \geq \alpha(G)^n$.
2. It is obvious that $\Theta(G_1 \otimes G_2) \geq \Theta(G_1)\Theta(G_2)$. However (see [Alo98], [Hae79]), there exist simple graphs $G_1 = (V_1, E_1)$ and $G_2 = (V_2, E_2)$, such that $\Theta(G_1 \otimes G_2) > \Theta(G_1)\Theta(G_2)$ (1)
3. For all $G \in \bigcup_{n=0}^\infty G(n)$, it holds by [Sha56], that $\Theta(G) = \alpha(G)$ (single letter).
4. For $G \in C_5$, it holds by [Lov79] that:
   $\Theta(G) = \sqrt[2]{\alpha(G^{2n})} = \sqrt{5}$ (multi letter).
5. Let $S = \{\{1\}\} \subseteq G$ and $G = S \sqcup C_5$. We know by Zuidam’s characterization in [Zui99] that $\Theta(G) = 1 + \sqrt{5}$ (see Theorem 27 and Remark 29), but there does not exists a $n \in \mathbb{N}_0$ such that $\Theta(G) = \sqrt[n]{\alpha(G^{2n})} = 1 + \sqrt{5}$.

Therefore, the limit is necessary in Theorem 5.

Remark 7. In his paper [Ahls70], Rudolf Ahlswede wrote “One would like to have a ‘reasonable’ formula for $C_0$, which does not “depend on an infinite product space.” Such a formula is unknown. An answer as: for given $d$ there exists a $k = k(d)$ such that $N(nk, 0) = N(k, 0)^d$ could be considered “reasonable.”

$N(n, 0)$ denotes the maximal $N$ for which a zero-error code for $n$ exists. Ahlswede made this comment in 1970, where the result of Lovász was not known. But by that result and point 5) in the remark it is clear, that such a result is not possible. Ahlswede’s question about a reasonable formula can be interpreted in the weakest form as a question about Turing computability.

A partial function from $\mathbb{N}_0$ to $\mathbb{N}_0$ is called partial recursive if it can be computed by a Turing machine; that is, if there exists a Turing machine that accepts input $x$ exactly when $f(x)$ is defined, in which case it leaves the string $f(x)$ on its tape upon acceptance.

We would like to make statements about the computability of the zero-error capacity. This capacity is generally a real number. Therefore, we first define when a real number is
computable. For this we need the following two definitions. We follow [Go30], [Go34], [Kle52], ...

**Definition 8.** A sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) is called a computable sequence if there exist recursive functions \( a, b, s : \mathbb{N}_0 \rightarrow \mathbb{N}_0 \) with \( b(n) \neq 0 \) for all \( n \in \mathbb{N}_0 \) and

\[
r_n = (-1)^s(n) \frac{a(n)}{b(n)}, \quad n \in \mathbb{N}_0.
\]

**Definition 9.** A real number \( x \) is said to be computable if there exists a computable sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) such that \( |x - r_n| < 2^{-n} \) for all \( n \in \mathbb{N}_0 \). We denote the set of computable real numbers by \( \mathbb{R}_c \).

We want to examine the zero-error capacity of general discrete memoryless channels for computability. As described above, each channel can be represented by a simple graph. We therefore examine the function \( i \), which was defined in the first chapter, for computability. We first define computable functions.

**Definition 10.** A function \( f : G \rightarrow \mathbb{R}_c \) is called computable if there exists a computable sequence of rational numbers \( \{r_n\}_{n \in \mathbb{N}} \) such that \( f \circ i : \mathbb{N}_0 \rightarrow \mathbb{R}_c \) is a computable function, that means there are recursive functions \( a, b, s, \phi : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0 \) with \( b(n, m) \neq 0 \) such that for all \( n \in \mathbb{N}_0 \) holds:

For all \( m \in \mathbb{N}_0 \) we have

\[
|f \circ i(n) - (-1)^{s(n,m)} \frac{a(n,m)}{b(n,m)}| < \frac{1}{2^m}.
\]

**Remark 11.** The above Definition 10 is equivalent to the following. There are four recursive functions \( a, b, s, \phi : \mathbb{N}_0^2 \rightarrow \mathbb{N}_0 \) with \( b(n,m) \neq 0 \) such that for all \( n, M \in \mathbb{N}_0 \), the following holds for all \( m \geq \phi(n, M) \):

\[
|f \circ i(n) - (-1)^{s(n,m)} \frac{a(n,m)}{b(n,m)}| < \frac{1}{2^M}.
\]

This definition is equivalent to Definition 10 in the sense that both approaches characterize the same class of functions.

**Observation 12.** The property \( f \) being computable does not depend on the choice of \( i : \mathbb{N}_0 \rightarrow G \), if \( i \) is bijective and recursive.

The problem is that although we have a representation for \( C_0(W) \) and \( \Theta(G) \) as the limit of a convergent sequence, we do not have an effective estimate of the rate of convergence as defined by Definition 9. So if we want to calculate the first bits of the binary representation of the number \( \Theta(G) \) even for a fixed graph \( G \), this is not possible with the result of Shannon alone. (Naturally, the same also applies to the first numbers of the decimal representation of the number \( \Theta(G) \).) An approach would now be, e.g. for the decimal representation, to derive the best possible lower bound for \( \Theta(G) \) from the achievability part and derive a good upper bound for \( \Theta(G) \) from an approach for the inverse part. If the two bounds match for the first \( L \) decimal places, then we have determined \( \Theta(G) \) for the first \( L \) decimal places. Today, even for a fixed graph \( G \), there is generally no algorithm that produces a computable sequence of computable numbers as a monotonically increasing sequence of lower bounds and a computable sequence of computable numbers as a monotonically decreasing sequence of upper bounds, both converging to \( \Theta(G) \). Indeed, it is unclear whether \( \Theta : G \rightarrow \mathbb{R}_c \) always applies. If this does not apply, then of course such an algorithm cannot exist. It should be noted here that in this discussion we do not require that the algorithm depends recursively on the graph \( G \), i.e. an individual algorithm can be developed for each graph \( G \). This is possible precisely if \( \Theta : G \rightarrow \mathbb{R} \) applies. This discussion is reflected in the level of knowledge about the behavior of the function \( \Theta \). It was a huge step forward that Lovász [Lov79] calculated \( \Theta(C_5) \) using the Lovász Theta function. Since then, \( \Theta(C_7) \) has not been determined [PS18]. In general, it is known that the Lovász Theta function is not tight [Hae79]. The requirement for the computability of a function \( f \) according to Definition 10 is of course much stronger than just the requirement for the property \( f \) : \( G \rightarrow \mathbb{R}_c \), because in Definition 10, the algorithm, i.e. the functions \( a \) and \( b \), depend of course, recursively on the graph.

### III. A TURING MACHINE WITH A KOLMOGOROV ORACLE

In the following we want to find an algorithm or Turing machine which can compute the zero-error capacity with the help of a Kolmogorov oracle. More specifically, we want to compute sharp lower and upper bounds for \( C_0 \) and \( \Theta \), whereby we can specify the approximation quality as desired. In this section we will introduce a Kolmogorov oracle. For the definition of a Turing machine we refer to [Soa87], [Wei00]. A Turing machine is a mathematical model of an abstract machine that manipulates symbols on a tape according to certain given rules. It can simulate any given algorithm and therewith provides a simple but very powerful model of computation. Turing machines have no limitations on computational complexity, have unlimited computing capacity and storage, and execute programs completely error-free. First we need the following definition.

**Definition 13.** A subset \( G_1 \subset G \) is called semi-decidable if there is a Turing machine \( TM_1 \) with the two states “stop” and “don’t stop”, such that \( TM_1(G) \) stops if and only if \( G \in G_1 \). Therefore \( TM_1 \) has only one stop state.

**Definition 14.** A subset \( G_1 \subset G \) is called decidable, if \( G_1 \) and \( G_1^c \) are semi-decidable.

\( G_1 \subset G \) is decidable if and only if there is a Turing machine \( TM_2 \) with the two states \{0, 1\} which gives \( TM_2(G) = 1 \) if and only if \( G \in G_1 \). The Turing machine stops for every input \( G \in G_1 \). Therefore the characteristic function of the set \( G_1 \) is described by the definition above. We have the following lemma.

**Lemma 15.** Let \( \lambda \in \mathbb{R}_c, \lambda \geq 0 \). Then the set \( \mathcal{G}(\lambda) := \{ G \in G : \Theta(G) > \lambda \} \) is semi-decidable.

The following lemma will show that \( G(\lambda) \) is recursively enumerable.

**Lemma 16.** Let \( \lambda \in \mathbb{R}_c, \lambda \geq 0 \). There exists a partial recursive function \( \phi_\lambda : \mathbb{N}_0 \rightarrow G \) with \( \phi_\lambda(\mathbb{N}_0) = G(\lambda) \) and
Let $f_m(G) = \frac{1}{2^m} \log_2 \Theta(G^{2^m})$ and $G(\lambda) = \{G \in \mathcal{G} : \Theta(G) > \lambda\}$. Let $i$ be the enumeration function of the graphs in $\mathcal{G}$ defined above. $G_i$ with $i \in \mathbb{N}_0$ is the $(i-1)$th graph in $\mathcal{G}$ concerning this enumeration. Let $\Delta$ be an empty set. Now we define the following algorithm on the Turing machine $T_M$:

1) Compute $f_1(G_1)$, if $f_1(G_1) > \lambda$, then $\phi_1(1) = G_1$, otherwise add 1 to $\Delta$.

2) Compute $f_1(G_2)$ and if $1 \in \Delta$ compute $f_2(G_1)$.

If $f_2(G_1) > \lambda$ then $\phi_1(1) = G_1$ and remove 1 from $\Delta$. If $f_1(G_2) > \lambda$ then $\phi_1(2) = G_2$, otherwise add 2 to $\Delta$.

3) Add $k$ to $\Delta$ and for all $j \in \Delta$:
   a) Compute $f_{k-j+1}(G_j)$.
   b) If $f_{k-j+1}(G_j) > \lambda$ then $\phi_1(j) = G_j$ and remove $j$ from $\Delta$.

The algorithm produces a recursive enumeration of the set $G(\lambda)$. Therefore, we need a specific enumeration for

- the set $\mathbb{N}_0$ and
- the set of the partial recursive functions.

The problem is that the natural listing of the set of natural numbers is inappropriate because many numbers in $\mathbb{N}_0$ are too large for the natural enumerations. We start with the set of partially recursive functions from $\mathbb{N}_0$ to $\mathbb{N}_0$. A listing $\Phi = \{\Phi_i : i \in \mathbb{N}_0\}$ of the partial recursive functions $f : \mathbb{N}_0 \to \mathbb{N}_0$ is called optimal listing if for any other recursive listing $\{g_i : i \in \mathbb{N}_0\}$ of the set of recursive functions there is a constant $C_1$ such that for all $i \in \mathbb{N}_0$ holds: There exists a $t(i) \in \mathbb{N}_0$ with $t(i) \leq C_1 i$ and $\Phi_{t(i)} = g_i$. This means that all partial recursive functions $f$ have a small Gödel number with respect to the system. Schnorr [Sch74] has shown that such an optimal recursive listing of the set of partial recursive functions exists. The same holds true for the sets of natural numbers $\mathbb{N}_0$.

For $\mathbb{N}_0$ let $u_{\mathbb{N}_0}$ be an optimal listing. For the set $\mathcal{F}$ of partially recursive functions, let $u_{\mathcal{F}}$ be an optimal listing. Then we define $C_{u_{\mathcal{F}}} : \mathcal{F} \to \mathbb{N}_0$ with $C_{u_{\mathcal{F}}}(f) := \min\{k : u_{\mathcal{F}}(k) = f\}$ and $C_{u_{\mathbb{N}_0}} : \mathbb{N}_0 \to \mathbb{N}_0$ with $C_{u_{\mathbb{N}_0}}(n) := \min\{k : u_{\mathbb{N}_0}(n) = n\}$ as the Kolmogorov complexity of $f$ and $n$ in terms of the optimal listings $u_{\mathcal{F}}$ and $u_{\mathbb{N}_0}$.

Furthermore, for the set $\mathcal{G}$ we define $C_{u_{\mathcal{G}}}(G) := \min\{k : i(u_{\mathbb{N}_0}(k)) = G\}$. This is the Kolmogorov complexity generated by $u_{\mathbb{N}_0}$ and $i$. On $\mathcal{F}$, $\mathbb{N}_0$ and $\mathcal{G}$ we introduce a new order criterion. We want to sort the elements of these sets in terms of Kolmogorov complexity:

$G_1 \leq_K G_2 \iff C_{u_{\mathcal{G}}}(G_1) \leq C_{u_{\mathcal{G}}}(G_2)$

For $\mathcal{F}$ and $\mathbb{N}_0$ we define this analogously.

**Definition 17.** The Kolmogorov oracle $O_{K,\mathcal{G}}(\cdot)$ is a function from $\mathbb{N}_0$ to the power set of the set of graphs that produces a list

$O_{K,\mathcal{G}}(n) := \{G : C_{u_{\mathcal{G}}}(G) \leq n\}$

for each $n \in \mathbb{N}_0$.

**Remark 18.** According to our definition of graphs and the set $\mathcal{G}$ with the listing $i$, this is the same as the listing $O_{K,\mathbb{N}_0}$ of the natural numbers $k$ with $k \leq n$.

Let $T_M$ be a Turing machine. We say that $T_M$ can use the oracle $O_{K,\mathcal{G}}$ if, for every $n \in \mathbb{N}_0$, on input $n$ the Turing machine gets the list $O_{K,\mathcal{G}}(n)$. With $T_M(O_{K,\mathcal{G}})$ we denote a Turing Machine that has access to the Oracle $O_{K,\mathcal{G}}$.

We have the following Theorem.

**Theorem 19.** Let $\lambda \in \mathbb{R}_-, \lambda > 0$. Then the set $G(\lambda)$ is decidable with a Turing machine $T_M^*(O_{K,\mathcal{G}})$. This means there exists a Turing machine $T_M^*(O_{K,\mathcal{G}})$, such that the set $G(\lambda)$ is computable with this Turing machine with the oracle.

**Corollary 20.** Let $\lambda \in \mathbb{R}_-, \lambda \geq 0$. Then, the set $\{G : \Theta(G) \leq \lambda\}$ is semi-decidable for Turing machines with oracle $O_{K,\mathbb{N}_0}(O_{K,\mathcal{G}})$.

**Remark 21.**

1) Noga Alon has asked if the set $\{G : \Theta(G) \leq \lambda\}$ is semi-decidable (see [AL06]). We give a positive answer to this question if we can include the oracle.

2) We do not know if $C_0$ is computable concerning $T_M(O_{K,\mathcal{G}})$.

Let $M \in \mathbb{N}_0$ be a number with $2^M \geq |\mathcal{V}(G)|$. We set $I_{0, M} = [0, \frac{1}{2^M}]$ and $I_{k, M} = \left[\frac{k}{2^M}, \frac{k+1}{2^M}\right]$ for $k = 1, 2, \ldots, 2^M - 1$. We have the following theorem.

**Theorem 22.** There exists a Turing machine $T_M^{(1)}(\cdot, O_{K,\mathbb{N}_0})$ with $T_M^{(1)}(\cdot, O_{K,\mathbb{N}_0}) : \mathcal{G} \to \{0, 1, \ldots, 2^M - 1\}$ such that for all $G \in \mathcal{G}$ with $|\mathcal{V}(G)| \leq 2^M$ holds

$T_M^{(1)}(G, O_{K,\mathbb{N}_0}) = r \iff \Theta(G) \in I_{r, M}$

Thus, this approach does not directly provide the computability of $C_0$ through $T_M^{(1)}$ with the oracle $O_{K,\mathbb{N}_0}$. However, we can compute $C_0$ with any given accuracy.

**Remark 23.** We have seen that in order to prove the computability of $C_0$ or $\Theta$ we need computable converses in the sense of Theorem 26. In this sense, the recent characterization of Zuiddam [Zui19] using the functions from the asymptotic spectrum of graphs is interesting. We will examine this approach with regard to predictability in the next section.

IV. CHARACTERIZATION OF THE SHANNON ZERO-ERROR CAPACITY

Shannon’s characterization of the zero-error capacity according to Theorem 5 can be interpreted as characterization over the achievable part of information theory. Of course this can not be interpreted as an effective, i.e. computable, characterization, since no effective estimate of the speed of convergence is known. Zuiddam has recently achieved, based on Strassen’s work, a very interesting characterization of Shannon’s zero-error capacity, which can be interpreted as a characterization by converse, i.e. a sharp upper bound. We will now examine Zuiddam’s representation in terms of its
The complement $u,v$:

Let further standard notions like graph homomorphism and graph preorder and introduce the asymptotic spectrum of graphs.

Theorem 27 (Zui19).

Theorem 26. The zero-error capacity $C_0$ and thus the function $\Theta$ is Turing computable if and only if there is a computable sequence $\{F_N\}_{N \in \mathbb{N}_0}$, so that the following conditions apply:

1. For all $N \in \mathbb{N}_0$, $F_N(\emptyset) = \Theta(\emptyset)$ and $F_N = \Theta(G)$ for all $G \in \mathcal{G}$.
2. $\lim_{N \to \infty} F_N(\emptyset) = \Theta(\emptyset)$ for all $G \in \mathcal{G}$.

Now we state the result of [Zui19], discuss the Strassen preorder and introduce the asymptotic spectrum of graphs. Finally, we show the decidability of the preorder with a Turing machine using an oracle. To state the result of [Zui19] we need further standard notions like graph homomorphism and graph complement. Let $G$ and $H$ be graphs. A graph homomorphism $f : G \to H$ is a map $f : V(G) \to V(H)$ such that for all $u,v \in V(G)$, if $(u,v) \in E(G)$, then $(f(u), f(v)) \in E(H)$. In other words, a graph homomorphism maps edges to edges. The complement $\overline{G}$ of $G$ is defined by $V(\overline{G}) = V(G)$ and $E(\overline{G}) = \{(u,v) : (u,v) \notin E(G), u \neq v\}$. For $G_1, G_2 \in \mathcal{G}$, let $G_1 \leq G_2$ if there exists a homomorphism from $G_1$ to $G_2$. Let furthermore $G_1 \leq G_2$ if there is a sequence $(x_N)_{N \in \mathbb{N}_0}$ such that $x_N/N \to 1$ when $N \to \infty$ such that for all $N \in \mathbb{N}_0$ we have $G_1 \leq (G_2^{\oplus N})^{1 \cup F_N}$. We call $\leq$ the asymptotic preorder induced by $\leq$.

Theorem 27 (Zui19). $\mathcal{G}$ is a collection of graphs which is closed under the disjoint union $\sqcup$ and the strong graph product $\otimes$, and which contains a graph with a single vertex, $K_1$. Define the asymptotic spectrum $X(\mathcal{G})$ as the set of all maps $\phi : \mathcal{G} \to \mathbb{R}_{\geq 0}$ such that, for all $G, H \in \mathcal{G}$:

1. If $G \leq H$, then $\phi(G) \leq \phi(H)$
2. $\phi(G \sqcup H) = \phi(G) + \phi(H)$
3. $\phi(G \otimes H) = \phi(G)\phi(H)$
4. $\phi(K_1) = 1$.

Then we have:

1. $G \leq H$ if and only if $\forall \phi \in X(H)$ $\phi(G) \leq \phi(H)$
2. $\Theta(G) = \min_{\phi \in X(G)} \phi(G)$.

Remark 28. The Theorem 27, especially point 2, is interesting with regard to the discussion in Remark 23. For example, if all $\phi \in X(G)$ have the property that they can be computed as functions $\phi : \mathcal{G} \to \mathbb{R}_{\geq 0}$ and if the set $X(\mathcal{G})$ is recursively enumerable, then we could immediately prove that for $G \in \mathcal{G}$, $\Theta(G) \in \mathbb{R}_{\geq 0}$ always applies, which has been still open up to now, as already mentioned. So far, however, it is both unclear whether $\phi \in X(\mathcal{G})$ always applies to $\phi : \mathcal{G} \to \mathbb{R}_{\geq 0}$ or whether all $\phi \in X(\mathcal{G})$ can be computed. The proof in [Zui19] from Theorem 27 is not constructive. The Zorn lemma is needed. Only a few functions from the asymptotic graph spectrum are known to date, e.g. Lovász Theta number, Fractional Haemers bound, and Fractional orthogonal rank. For these functions (see [Zui19]) it is not clear whether they always fulfill $\phi : \mathcal{G} \to \mathbb{R}_{\geq 0}$, because these are defined by the sequence of suitable functions. It is not clear whether effective convergence occurs here even for fixed $G$.

Remark 29. The characterization 2. from Theorem 27 leads directly to the following property of the $\Theta$ function. For any two graphs $G_1, G_2$ we have $\Theta(G_1 \otimes G_2) = \Theta(G_1) \Theta(G_2)$ and if and only if $\Theta(G_1 \sqcup G_2) = \Theta(G_1) + \Theta(G_2)$ applies. Consequently, we have $\Theta(S \sqcup C_5) = 1 + \sqrt{5}$. So the answer to Ahlswede’s question in Remark 7 for $d \leq 5$ is positive, but negative for $d > 5$. It is interesting that $k(5) = 2$.

Now we are prepared to prove the decidability of the preorder with an oracle. We have $G \leq H \iff \forall \phi \in X(G)$ holds $\phi(G) \leq \phi(H)$.

$X(\mathcal{G})$ is a term for an infinite number of functions $\phi$ and it is not clear which $\phi \in X(G)$ are computable. Furthermore, $\leq$ is a binary relation on $\mathcal{G} \times \mathcal{G}$, the set of pairs of graphs. Is this binary relation Turing computable? That means there exists a Turing machine $TM$ with $TM : G \times G \to \{0, 1\}$ with $TM((G, H)) = 1 \iff G \leq H$.

Our goal is to use a powerful oracle, such that with the help of the oracle there exists a Turing machine which computes the binary relation $\leq$.

Definition 30. Let $\phi_k, k \in \mathbb{N}_0$, be the list of partial recursive functions. $\phi_k$ is called total if the domain of $\phi_k$ equals $\mathbb{N}_0$.

Definition 31. Let $T_{\text{tot}} = \{k \in \mathbb{N}_0 : \phi_k$ is a total function$\}$. Then $O_{T_{\text{tot}}}$ is defined as the following oracle. In the calculation step $l$, $TM$ asks the oracle if $k \in T_{\text{tot}}$ is satisfied. $TM$ receives in a calculation step the answer yes or no. The Turing machine uses this answer for the next computation, etc. New queries can always be made to the oracle.

Theorem 32. $\leq$ is a binary operation decidable by a Turing machine $TM(\cdot, O_{T_{\text{tot}}})$.

The $O_{T_{\text{tot}}}$ Oracle is a very strong one. It would be desirable to show the theorem using a weaker oracle.

A detailed version of the paper with all proofs and definitions can be found in [BoDe20].
REFERENCES

[Ahls70] Rudolf Ahlswede. A note on the existence of the weak capacity for channels with arbitrarily varying channel probability functions and its relation to Shannon’s zero-error capacity. *Ann. Math. Stat.*, Vol. 41, No. 3, 1027-1033, 1970.

[Alo98] Noga Alon. The Shannon capacity of a union. *Combinatorica*, 18(3):301–310, 1998.

[AL06] N. Alon and E. Lubetzky. The Shannon capacity of a graph and the independence numbers of its powers, *IEEE Trans. on Information Theory* 52, 2172-2176, 2006.

[BoDe20] Holger Boche and Christian Deppe. Computability of the Zero-Error capacity with Kolmogorov Oracle. *Arxiv*, 2020.

[Go30] K. Gödel. Die Vollständigkeit der Axiome des logischen Funktionskalküls. *Monatshfte für Mathematik*, vol. 37, no. 1, pp. 349–360, 1930.

[Go34] K. Gödel. On undecidable propositions of formal mathematical systems. *Notes by Stephen C. Kleene and Barkely Rosser on Lectures at the Institute for Advanced Study*, Princeton, NJ, 1934.

[Kle52] S.C. Kleene. Introduction to Metamathematics. *Van Nostrand*, New York: Wolters-Noordhoff, 1952.

[Mi61] M. Minsky. Recursive unsolvability of Post’s problem of ‘tag’ and other topics in theory of Turing machines. *Ann. Math.*, vol. 74, no. 3, pp. 437–455, 1961.

[Hae79] Willem Haemers. On some problems of Lovász concerning the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(2):231–232, 1979.

[KO98] J. Körner and A. Orlitsky. Zero-error information theory. *IEEE Trans. Inform. Theory*, 44(6):2207–2229, 1998.

[Lov79] László Lovász. On the Shannon capacity of a graph. *IEEE Trans. Inform. Theory*, 25(1):1–7, 1979.

[PS18] S. Polak and A. Schrijver. New lower bound on the Shannon capacity of C7 from circular graphs. *Information Processing Letters* 143, 2018.

[PoRi17] M. B. Pour-El and J. I. Richards. Computability in Analysis and Physics. *Cambridge University Press*, Cambridge, 2017.

[Sch74] Claus Peter Schnorr. Optimal enumerations and optimal Gödel numberings. *Mathematical Systems Theory* 8(2), pp. 181–191, 1974.

[Sha56] Claude E. Shannon. The zero-error capacity of a noisy channel. *Transactions on Information Theory*, IT-2(September):8–19, 1956.

[Soa87] R.I. Soare. Recursively Enumerable Sets and Degrees. Berlin, Heidelberg: Springer-Verlag, 1987

[Spe49] E. Specker. Nicht konstruktiv beweisbare Sätze der Analysis, *Journal of Symbolic Logic*, vol. 14, no. 3, pp. 145–158, Sep. 1949.

[Str86] Volker Strassen. The Asymptotic Spectrum of Tensors and the Exponent of Matrix Multiplication. In *Proceedings of the 27th Annual Symposium on Foundations of Computer Science*, SFCS ’86, pages 49–54, Washington, DC, USA, 1986. IEEE Computer Society.

[Str87] Volker Strassen. Relative bilinear complexity and matrix multiplication. *J. Reine Angew. Math.*, 375/376:406–443, 1987.

[Str88] Volker Strassen. The asymptotic spectrum of tensors. *J. Reine Angew. Math.*, 384:102–152, 1988.

[Str91] Volker Strassen. Degeneration and complexity of bilinear maps: some asymptotic spectra. *J. Reine Angew. Math.*, 413:127–180, 1991.

[Tur36] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem, *Proc. London Math. Soc.*, vol.2, no. 42, pp. 230–265, 1936.

[Tur37] Alan M. Turing. On computable numbers, with an application to the Entscheidungsproblem. A correction, *Proc. Lon-don Math. Soc.*, vol. 2, no. 43, pp. 544–546, 1937.

[Vra19] P. Vrana. Probabilistic refinement of the asymptotic spectrum of graphs, CoRR abs/1903.0185, 2019.

[Wei00] Klaus Weihrauch, Computable Analysis - An Introduction, Springer-Verlag, Berlin Heidelberg, 2000

[Zui18] Jeroen Zuiddam. Asymptotic spectra, algebraic complexity and moment polytopes. PhD thesis, University of Amsterdam, 2018.

[Zui19] Jeroen Zuiddam. The asymptotic spectrum of graphs and the Shannon capacity. *J. Combinatorica*, 2019.