Uncomputability of the generalized capacity

David Elkouss, Member, IEEE, and David Pérez-García

Abstract—What is the maximum rate at which we can send information over a channel? We define the capacity of a channel as the answer to this question. For an important family of channels the capacity is given by a simple optimization problem as proven in Shannon’s noisy coding theorem. Furthermore, for these channels, we have the Blahut-Arimoto algorithm that allows to efficiently solve the optimization problem and compute the capacity. In groundbreaking work, Verdu and Han proved a coding theorem for general channels. However, despite considerable effort, there is no equivalent to the Blahut-Arimoto algorithm for computing the generalized capacity. Here, we show that such an algorithm cannot exist.

I. INTRODUCTION

The theory of communications began to take shape with the contributions among others of Hartley [1] and Nyquist [2], [3]. However, it was mainly Shannon [4] that single handedly established the foundations of the field in the mid forties. A communications theory should deal with the abstract problem of transmitting messages stripped of any signification. A message is just an element from a set of messages and “...the fundamental problem of communication is that of reproducing at one point either exactly or approximately a message selected at another point.”

These distant points or parties are connected through a transmission line that Shannon abstracted into a channel, a black box that takes letters from an alphabet at the input and outputs letters from a possibly different alphabet. Closely related with the concept of transmitting messages through a channel is the concept of mapping the message to a sequence of inputs of the channel or encoding the message. The ratio between the logarithm of the set of messages and the length of the sequence is called the rate of the encoding. At the other end of the channel, the other party can apply an inverse mapping that takes sequences of letters from the output alphabet to the set of messages. A message is correctly transmitted if the composition of the encoding, the action of the channel and the decoding acting on the message behave like the identity function.

The natural question that Shannon posed is: What is the maximum rate that can be achieved with an arbitrarily small error? In an ingenuity tour de force, he proved that in the case of memoryless channels the answer to this question exists. This number is the capacity of the channel.

Following Shannon’s work, more and more general classes of channels were successfully dealt with [5], [6], [7], [8] culminating with the generalized capacity formula [9]. In this last work, Verdu and Han derived a generalization of Shannon’s formula which essentially makes no assumption regarding the structure of the channel.

A coding theorem might furnish a formula for the capacity but it does not have any implications regarding its computability. That is, the existence of an algorithm that on input the channel outputs its capacity. Fortunately, for memoryless channels, the capacity is computable via the Blahut-Arimoto [10], [11] algorithm. Again for more general channels the situation is more complicated. In some concrete cases such as Gaussian channels with inter-symbol interference [12] or Gilber-Elliott channels [13], [14] the capacity can be computed, and recently the Blahut-Arimoto algorithm has been generalized to address finite state machine channels (FSMC) that verify certain properties [15], [16]. However no general algorithm is known that can address the full class of channels for which we have the generalized capacity formula [9]. Here we show that such a formula can not exist. More precisely, our main result is the following:

Theorem 1. Let \((n, m)\) be a duple of integers that is equal or pointwise larger than \((10, 62)\). Given a computable number \(\lambda \in (0, 1]\) (for instance a rational number in \((0, 1])\) and a FSMC \(N\) with input alphabet cardinality \(n\) and \(m\) states there is no algorithm that can decide the following yes/no questions:

\[
\begin{align*}
C(N) &< \lambda \\
C(N) &\geq \lambda
\end{align*}
\]

Let \((n, m)\) be a duple of integers that is equal or pointwise larger than \((8, 27)\). Given a computable number \(\lambda \in (0, 1)\) (for instance a rational number in \((0, 1])\) and a FSMC \(N\) with input alphabet cardinality \(n\) and \(m\) states there is no algorithm that can decide the following yes/no questions:

\[
\begin{align*}
C(N) &> \lambda \\
C(N) &\leq \lambda
\end{align*}
\]

The plan for the rest of the text is as follows. We introduce the notation in Section [II] and review some definitions regarding information theoretic quantities and capacities of discrete channels in Section [III]. Section [IV] is dedicated to probabilistic finite automata (PFA) and decision problems for PFA. We present our channel construction and prove the undecidability of its capacity in Section [V]. We finish the paper in Section [VI].
with a brief discussion of the implications of our result and some open questions.

II. NOTATION

We denote random variables by capital letters $X, Y, \ldots$, discrete alphabets by calligraphic capital letters $\mathcal{X}, \mathcal{Y}, \ldots$, channels, sources and PDFs by capital bold face letters $\mathbf{X}, \mathbf{Y}, \ldots$, and instances of random variables by lower case letters $x, y, \ldots$. We denote vectors with the same convention, whenever confusion might arise a superscript indicates the number of components of the vector and a subscript the concrete component: $X^n = (X_1, X_2, \ldots, X_n)$ or $x^n = (x_1, x_2, \ldots, x_n)$. We indicate a consecutive subset of $n$ components of the vector with subscript notation $[a, a + n - 1]$: $x_{[a, a+n-1]} = (x_a, x_{a+1}, \ldots, x_{a+n-2}, x_{a+n-1})$.

A vector is called a probability vector if all its entries are non-negative and add up to one. A matrix is called a stochastic matrix if all its columns are probability vectors. A stochastic matrix takes probability vectors to probability vectors.

III. CAPACITY OF DISCRETE CHANNELS

A. Definitions

In this paper we will focus on discrete channels. A discrete channel $W$ is defined by two discrete alphabets $\mathcal{X}$ and $\mathcal{Y}$ and a sequence of conditional probability distributions $\{W^n\}$:

$$W = \{W^n(y^n|x^n) = P(y^n|x^n), \forall x^n \in \mathcal{X}^n, y^n \in \mathcal{Y}^n\}_{n=1}^\infty$$

The task that we consider here is the transmission of information through a discrete channel. The transmission takes place in the following way. Let $\mathcal{M} = \{1, \ldots, M\}$ be the set of messages, $E : \mathcal{M} \rightarrow \mathcal{X}^n$ is an encoding function that takes a message to an input into $n$ uses of the channel, and $D : \mathcal{Y}^n \rightarrow \mathcal{M}$ is a decoding function that takes $n$ outputs of the channel to a message.

When a transmitter wants to send a message, he encodes it with $E$ and sends it through the channel, then the receiver applies $D$ to the received word and outputs a message. The decoding error is defined as:

$$\epsilon = \frac{1}{M} \sum_{i=1}^M P[D(W(E(i))) \neq i]$$

where we assume that all messages are chosen uniformly at random. We denote by a triple $(M, n, \epsilon)$ a code with $M$ codewords for which it exists an encoding function into $n$ channel uses and a decoding function that achieves an error $\epsilon$. The coding rate is defined as:

$$r = \frac{\log M}{n}$$

A rate $R$ is called $\epsilon$-achievable if for all $\delta > 0$ there exists $n_0$ such that for all $n > n_0$ there exists an $(M, n, \epsilon)$ code with rate:

$$r \geq R - \delta$$

The capacity of a channel is the supremum of the rates that are $\epsilon$-achievable for all $\epsilon \in (0, 1)$.

B. Formula for memoryless channels

$W$ is a discrete memoryless channel (DMC) if there exists some conditional probability distribution $W(y|x)$ such that for all $n$:

$$W^n(y^n|x^n) = \prod_{i=1}^n W(y_i|x_i)$$

where $x_n = (x_1, \ldots, x_n)$ and $y_n = (y_1, \ldots, y_n)$ are respectively a sequence of inputs and outputs of the channel.

Consider two random variables $X, Y$ with joint distribution $p(x, y)$, the information spectrum is the distribution of the random variable $i_{X,Y}$ given by:

$$i_{X,Y}(x, y) = \log \frac{p(y|x)}{p(y)}$$

The mutual information is the expected value of the information spectrum:

$$I(X; Y) = \langle i_{X,Y}(X, Y) \rangle_{XY}$$

The capacity of a DMC $W$ is given by the maximum over all input probability distributions of the mutual information between the input and the output of the channel $\mathcal{W}$:

$$C(W) = \sup_X I(X; Y)$$

C. Formula for general channels

Now we consider a situation more general than (9) where $W^n$ is not necessarily the $n$-th product of some $W$. First we define a discrete source analogously to a discrete channel in $\mathcal{X}$:

$$X = \{X^n(x^n) = P(x^n), \forall x^n \in \mathcal{X}^n\}_{n=1}^\infty$$

Let $\{Z_n\}_{n=1}^\infty$ be a sequence of random variables, the lim-inf and lim-sup in probability of the sequence are defined as:

$$\liminf_{n \to \infty} Z_n = \sup \{\alpha : \lim_{n \to \infty} P[Z_n \leq \alpha] = 0\}$$

$$\limsup_{n \to \infty} Z_n = \inf \{\alpha : \lim_{n \to \infty} P[Z_n \geq \alpha] = 0\}$$

Let $W$ be a discrete channel and let $Y$ denote the discrete source that is obtained by inputting $X$ through the channel $W$. Then the capacity of $W$ is given by $\mathcal{W}$:

$$C(W) = \sup_X I(X; Y)$$

$I(X; Y)$ denotes the inf-information rate between the two sources $X$ and $Y$. It is defined as the lim-inf in probability of the normalized information spectrum of the sources:

$$I(X; Y) = \liminf_{n \to \infty} \frac{1}{n} I(X^n; Y^n)$$

A useful upper bound on the inf-information rate is given by:

$$I(X; Y) \leq \liminf_{n \to \infty} \frac{1}{n} I(X^n; Y^n)$$

with the usual definition of limit inferior of a sequence.
D. Finite State Machine Channels

A channel can depend on past inputs and outcomes in very complicated ways. In order to attack the computability of the generalized capacity we are going to concentrate in the computability of a subset of discrete channels. More precisely, we are going to focus on FSMC which is the set of discrete channels that have its behavior dictated by a finite state machine [12].

A general channel can act differently depending on the previous sequence of inputs and outputs. That is, the sequence of past inputs and outputs defines a state and the channel action is a function of the state. A FSMC as a discrete channel has finite input and output alphabets but, in addition, also the set of possible states, that we denote by S, is required to be finite. Such a channel is completely characterized by the time-invariant conditional probabilities \( p(y, b|x, a) \) for all states \( a, b \in S \), input symbols \( x \in \mathcal{X} \) and output symbols \( y \in \mathcal{Y} \). These conditional probabilities denote the probability that the channel outputs the symbol \( y \) and transitions to the state \( b \) given that the channel is in state \( a \) and receives the input symbol \( x \). Then, given some initial state \( s_0 \), we can define \( W^n_{s_0} \) a sequence of probability distributions that characterize a discrete channel \( \mathbf{W}_{s_0} \) in the following way:

\[
W^n_{s_0}(y^n|x^n) = \sum_{s_n} W^n_{s_0}(y^n, s_n|x^n) \tag{20}
\]

where

\[
W^n_{s_0}(y^n, s_n|x^n) = \sum_{s_{n-1}} p(y_n, s_n|x_n s_{n-1}) W^{n-1}_{s_0}(y_{n-1} s_{n-1}|x_{n-1}) \tag{21}
\]

Analogously we can define a sequence of probability distributions to characterize the state of the channel:

\[
W^n_{s_0}(s_n|x^n) = \sum_{y^n} W^n_{s_0}(y^n, s_n|x^n) \tag{22}
\]

Note that we have used, abusing the notation, \( W^n_{s_0} \) both to define the conditional probability of the output and the state. It will be clear from the context to which one we refer.

IV. Probabilistic Finite Automata

A PFA \( A \) is given by a tuple \( A = (Q, \mathcal{W}, \mathcal{X}, v, F) \). \( Q \) denotes a finite set of states, \( \mathcal{W} \) denotes a finite input alphabet, \( \mathcal{X} \) denotes a finite set of stochastic matrices with cardinality equal to the cardinality of the input alphabet, \( v \) denotes an initial probability distribution over \( Q \) and \( F \subseteq Q \) denotes a set of accepting states.

The action of a PFA is defined by the transition probabilities from one state to another as a function of the input symbols. If the automaton is in the state \( q_a \) and reads the letter \( w \) it transitions to the state \( q_b \) with probability:

\[
p\left[q_a \rightarrow q_b|w\right] = (X_w)_{q_b, q_a} \tag{23}
\]

where we denote by \( \langle a, b \rangle \) the scalar product between vectors \( a \) and \( b \) and by \( \pi_{\mathcal{X}} \) a vector with ones in the positions indicated by \( \mathcal{X} \) and zeroes in the remaining positions. We exploit the same notation for the probability that the automaton transitions from the state \( q_a \) to the state \( q_b \) after reading the word \( w = (w_1, \ldots, w_{|w|}) \in \mathcal{W}^{|w|} \):

\[
p\left[q_a \rightarrow q_b|w\right] = \langle \pi_{\{q_a\}}, X_{w|w|} \cdots X_{w|v|} \pi_{\{q_b\}} \rangle \tag{25}
\]

More generally, if we have a probability distribution over the states given by the column vector \( x \) and the PFA reads the letter \( w \) then the new distribution over the states is given by \( X_{w|x} \). A particularly relevant probability is the probability that the automaton ends in an accepting state after reading some word \( w \). We call this probability the probability of accepting \( w \) or the value of \( w \). It can be computed

\[
\text{val}(A, w) = \langle \pi_{\mathcal{F}}, X_{w|w|} \cdots X_{w|v|} \rangle \tag{26}
\]

We call the value of \( A \), which we denote by \( \text{val}_{A} \), the supremum of the acceptance probabilities over all input words:

\[
\text{val}_{A} = \sup_{w \in \mathcal{W}^*} \text{val}(A, w) \tag{27}
\]

Whenever possible, we will represent graphically the different automata constructions. We will follow the following conventions. A state is denoted by a circle. An accepting state is denoted by a circle with a double line around it. In all automata in this paper the initial distribution will have one coefficient with weight one. We indicate the corresponding state with an arrow that does not come from any state.

We indicate with \( \rightarrow \rightarrow \) that if the automaton reads the letter \( w \) it transitions from the origin of the arrow to the state pointed by the arrow with probability \( p \). In order to avoid clutter we simplify the notation in several cases. If we do not show transitions corresponding with all input symbols, the missing transitions correspond to self-loops with probability one. We drop the probability and just write \( \rightarrow \rightarrow \) if a transition occurs with probability one. We drop the input symbol and just write \( p \rightarrow \rightarrow \) if all input symbols transition with the same probability.

Example 1. Consider the PFA given in Figure 1. The automaton in the figure has three states \( Q = \{q_1, q_2, q_3\} \), two input symbols \( \mathcal{W} = \{a, b\} \), the initial state is \( q_1 \) and there is a single accepting state \( q_3 \). By looking at the figure we can construct the stochastic matrices:

\[
X_a = \begin{pmatrix}
0.5 & 1 & 0 \\
0.5 & 0 & 0.5 \\
0 & 0 & 0.5
\end{pmatrix}, \quad X_b = \begin{pmatrix}
0 & 0 & 0 \\
0 & 1 & 0.5 \\
1 & 0 & 0.5
\end{pmatrix} \tag{28}
\]

Now, assume that we see the word \( w = baa \), we can easily compute its value:

\[
(0 & 0 & 1) X_b \cdot X_a \cdot X_a \cdot X_a \begin{pmatrix}
0 \\
0 \\
1
\end{pmatrix} = 0.25 \tag{29}
\]

A. Decision problems

Let us now consider several decision problems associated with PFA. First, let \( L_{A > \lambda} \) be the set of words with acceptance probability strictly higher than \( \lambda \). That is:

\[
L_{A > \lambda} = \{ w : \text{val}(A, w) > \lambda \}
\]
val(\mathcal{A}, w) > \lambda$. Is this set empty? This problem, known as the emptiness problem, was proved undecidable in \cite{18, 19, 20}. Recently, new proofs with explicit bounds in the number of states and the cardinality of the alphabet have been derived in \cite{21, 22, 23} together with an undecidability proof of several related sets.

**Lemma 1.** Let \((n, m)\) be a duple of integers that is equal or pointwise larger than \((2, 27)\). Given a computable \(\lambda \in (0, 1)\) (for instance a rational in \((0, 1)\)) and a PFA with alphabet size \(n\) and \(m\) states the emptiness of \(L_{A > \lambda}\) is undecidable.

A related problem is the isolation threshold problem. A threshold \(\lambda\) is said to be isolated if there exists \(\epsilon > 0\) such that for all \(w \in \mathcal{W}\) the word \(w\) is accepted with probability separated at least \(\epsilon\) from the threshold: \(|\text{val}(\mathcal{A}, w) - \lambda| > \epsilon\). Given an automata \(\mathcal{A}\) and \(\lambda\), the isolated threshold problem is to decide whether or not \(\lambda\) is an isolated threshold. It was first proved undecidable for \(\lambda \in (0, 1)\) in \cite{24, 25}. A simpler proof can be found in \cite{21}.

We introduce a third problem that we call the value problem. Is \(\text{val}_A\) strictly smaller than some \(\lambda\)? Note that this problem is neither equivalent to the emptiness of \(L_{A > \lambda}\) nor to the emptiness of \(\{w : \text{val}(\mathcal{A}, w) \geq \lambda\}\). However, it is easy to see that \(\text{val}_A < \lambda\) if and only if both \(L_{A > \lambda}\) is empty and \(\lambda\) is an isolated threshold. In the particular case that \(\lambda = 1\), \(L_{A > \lambda}\) is trivially empty and then \(\text{val}_A < 1\) is equivalent to the isolated threshold problem for \(\lambda = 1\). Although the undecidability of the isolated threshold in the range \((0, 1)\) was known from the seventies \cite{24, 25}, it was only recently \cite{23} that it was proved for \(\lambda = 1\). In that proof, no explicit bounds on the number of states and cardinality of the alphabet were computed. We reprove the result in the appendix deriving concrete bounds for both values. Building on top of this construction we also show that the value problem is undecidable for any \(\lambda \in (0, 1]\).

**Theorem 2.** Let \((n, m)\) be a duple of integers that is equal or pointwise larger than \((3, 62)\). Given a computable \(\lambda \in (0, 1]\) (for instance a rational in \((0, 1]\)) and a PFA with alphabet size \(n\) and \(m\) states the value problem is undecidable.

**B. Stable and resettable PFAs**

We call a PFA a stable PFA if one of the transition matrices is equal to the identity matrix \(X_{id}\). The reason is that for such a PFA reading the symbol corresponding to the identity leaves the state probabilities unchanged. Let \(u\) be any probability vector, then

\[
u = X_{id}u	ag{30}\]

We call a PFA a resettable PFA if one of the transition matrices, \(X_n\), takes the state back to the initial state. Let \(u\) be any probability vector, then

\[
v = X_nu	ag{31}\]

Let \(\gamma\) be a map from PFAs to PFAs such that given a PFA \(\mathcal{A}, \gamma(\mathcal{A})\) is a PFA with an extended input alphabet composed of the original alphabet together with the additional symbols id and rt and the corresponding matrices \(X_{id}\) and \(X_n\) as given by (30) and (31). That is, \(\gamma\) transforms any PFA into a stable and resettable PFA.

As we prove formally in the appendix, the emptiness and value decision problems remain undecidable for stable and resettable PFAs.

**Lemma 2.** Let \(\lambda \in (0, 1)\), \((n, m)\) be a duple of integers that is equal or pointwise larger than \((4, 27)\). The emptiness problem is undecidable for stable and resettable PFA.

Let \(\lambda \in (0, 1)\), \((n, m)\) be a duple of integers that is equal or pointwise larger than \((5, 62)\). The value problem is undecidable for stable and resettable PFA.

V. THE CAPACITY OF A FAMILY OF CHANNELS BASED ON PFA

Given a stable and resettable PFA \(\mathcal{A}\) we define the channel \(V_{\mathcal{A}}\). The input alphabet of the channel takes values in \(\{0, 1\} \times \mathcal{W}\), which we identify with two different inputs: a data input and a control input. The data input is transmitted to the output; noiselessly if \(\mathcal{A}\) is in an accepting state or, if \(\mathcal{A}\) in any other state, the channel outputs uniformly at random an element of the output alphabet. More concretely, the output of the channel, which is independent of the control input, is defined by the following conditional probability:

\[
p(y_n|x_n, s_{n-1}) = \begin{cases} 
1 & \text{if } s_{n-1} \notin \mathcal{F} \\
\frac{1}{2} & \text{if } s_{n-1} \in \mathcal{F} \text{ and } y_n = x_n \\
0 & \text{else}
\end{cases}
\tag{32}\]

The control input is fed to \(\mathcal{A}\), which begins in the initial state, and the state transition probabilities are dictated by the PFA:

\[
p(s_n|x_n, s_{n-1}) = \langle \pi_{x_n}, X_{c_n, \gamma s_{n-1}} \rangle.	ag{33}\]

**Lemma 3.** \(C(V_{\mathcal{A}}) > \lambda \iff L_{A > \lambda} \text{ is not empty.}\)

**Proof:** \(\Leftarrow\) Let \(w\) be a word such that \(\text{val}(\mathcal{A}, w) > \lambda\), then there exists some \(\delta > 0\) such that \(\text{val}(\mathcal{A}, w) = \lambda + \delta\), furthermore let \(|w| = m\). Consider the following protocol, the input into the control register is the deterministic sequence \((c_i)_{i=1}\) with

**Fig. 1:** Automaton with three states \(Q = \{q_1, q_2, q_3\}\), two input symbols \(W = \{a, b\}\), the initial state is \(q_1\) and there is a single accepting state \(q_3\).
Finally by choosing the first \( -2 \phi \) sequence, we can bound the conditional entropy of the output given the input as follows:

\[
\sum_{i=1}^{m+n} H(Y_i | Y_{[i-1]} X_{[1,m+n]} C_{[1,m+n]}) \leq m + H(Y_{[m+1,m+n]} | Y_{[1,m]} X_{[1,m+n]} C_{[1,m+n]}) \leq m + H(Y_{[m+1,m+n]} | X_{[1,m+n]} C_{[1,m+n]}) \leq m + 1 + (1 - \lambda - \delta)n
\]

where (35) follows by the chain rule, the inequality (36) by bounding the entropy of the first \( m \) uses by \( m \), the inequality (37) by removing the conditioning on \( Y_{[1,m]} \) and (38) holds from bounding the conditional entropy by (42) that we prove below.

After the first \( m \) uses, the automaton behaves like a noiseless channel with probability \( \lambda + \delta \) and like a completely random channel with the complementary probability. In consequence, we can bound the conditional entropy of the output of the \( m \) first uses by the following:

\[
H(Y_{[m+1,m+n]} | X_{[1,m+n]} C_{[1,m+n]}) = H((\lambda + \delta + (1 - \lambda - \delta)2^{-n})\phi + (1 - \lambda - \delta)(1 - 2^{-n})\rho) \\
= h(\lambda + \delta + (1 - \lambda - \delta)2^{-n}) + (1 - \lambda - \delta)(1 - 2^{-n}) \log(2^n - 1) \\
\leq 1 + (1 - \lambda - \delta)n
\]

where \( \phi = (1, 0, \ldots, 0) \) is a completely deterministic probability vector of length \( 2^n \), \( \rho = \left(0, \frac{1}{2^n-1}, \frac{1}{2^n-1}, \ldots, \frac{1}{2^n-1}\right) \) the maximally entropic vector of length \( 2^n - 1 \) and \( h(\epsilon) := -\epsilon \log \epsilon - (1 - \epsilon) \log(1 - \epsilon) \) is the binary entropy function.

Now we can use (38) to bound the mutual information of the first \( m+n \) uses:

\[
I(Y^{m+n}; X^{m+n}C^{m+n}) = H(Y^{m+n}) - H(Y^{m+n}|X^{m+n}C^{m+n}) = m + n - H(Y^{m+n}|X^{m+n}C^{m+n}) \geq n(\lambda + \delta) - 1
\]

Finally by choosing \( n \) larger than \((m\lambda + 1)/\delta\) we get \( I(Y^{m+n}; X^{m+n}C^{m+n}) > (m+n)\lambda \) which means that a rate larger than \( \lambda \) is achievable.

\( \Rightarrow \) We prove the second part of the theorem by the contrapositive. Suppose that \( L_{A>\lambda} \) is empty, then \( \text{val}_A \leq \lambda \).

Let \( S_i \) denote the state of the PFA at use \( i \), since the output only depends on the control input through the PFA state we have that \( H(Y_i | X_i C_{[1,i-1]}) \geq H(Y_i | X_i S_{i-1}) \). In consequence, we can bound from below the conditional entropy of the output given the input as follows:

\[
H(Y^n | X^n C^n) = \sum_{i=1}^{n} H(Y_i | Y_{[i-1]} X^n C^n) \geq \sum_{i=1}^{n} H(Y_i | Y_{[i-1]} X_i C_{[1,i-2]}) \geq \sum_{i=1}^{n} p(S_{i-1} \in F) H(Y_i | S_{i-1} \in F, X_i = 0) + p(S_{i-1} \notin F) H(Y_i | S_{i-1} \notin F, X_i = 0) \geq n(1 - \lambda)
\]

Finally, we can plug the bound on the conditional entropy to obtain the desired result:

\[
-\frac{1}{n} I(Y^n; X^n C^n) = \frac{1}{n} (H(Y^n) - H(Y^n | X^n C^n)) \leq 1 - \frac{1}{n} H(Y^n | X^n C^n) \leq \lambda
\]

\( \Rightarrow \) Assume that \( \text{val}_A > \lambda \) then there exists \( w \) such that \( \text{val}(A, w) > \lambda \) and \( L_{A>\lambda} \) is not empty. Hence by Lemma 3 \( C(V_A) > \lambda \).

\( \Leftarrow \) Assume that \( C(V_A) > \lambda \), then by Lemma 3 \( L_{A>\lambda} \) is not empty which implies that \( \text{val}_A > \lambda \). 

**Corollary 2.** The capacity of \( V_A \) is given by:

\[
C(V_A) = \text{val}_A
\]

We prove now Theorem 1. The proof follows from Corollary 1, Lemma 3 and Lemma 2.

**Proof:**

The argument for the first case relies on the value problem. Suppose that there exists an algorithm that on input any channel \( \mathcal{N} \) it outputs whether or not the capacity is strictly smaller than \( \lambda \). Now, consider some channel \( V_A \). Then by Corollary 1 \( C(V_A) < \lambda \) if and only if \( \text{val}_A < \lambda \). This implies that the algorithm would also allow to decide the value problem, which is not possible since this is an undecidable problem for resettable and stable automata by Lemma 2.

The argument for the second case is very similar but relies on the emptiness problem. Suppose that there exists an algorithm that on input any channel \( \mathcal{N} \) it outputs whether or
not the capacity is strictly greater than $\lambda$. Now, consider some channel $V_A$. Then by Lemma 3 $C(V_A) > \lambda$ if and only if $L_{A>\lambda}$ is not empty. This implies that the algorithm would also allow to decide the emptiness of $L_{A>\lambda}$, which is not possible since this is an undecidable problem for resettable and stable automata by Lemma 2.

VI. Discussion

We have proved that no Blahut-Arimoto algorithm for the generalized capacity can exist. The proof technique could be extended to the capacities of quantum channels with memory. This proof connects with some recent results regarding the different capacities of memoryless quantum channels [26], [27], these results show some evidence that these capacities might be undecidable. Also memoryless zero error capacities, both classical and quantum are known to have highly non-trivial behaviour [28], [29], [30], [31], [32]. Are any of these quantities undecidable? The techniques used here exploit directly the memory of the channel. Hence, it is unlikely that a similar proof would allow to show undecidability of a memoryless quantity. On the other hand, it might be possible to extend our results to the zero-error capacities of channels with memory.

Verdu and Han proved “a completely general formula for channel capacity, which does not require any assumptions such as memorylessness, information stability, causality, etc.” [9]. The price to pay for such a formula is computability. Does our result render hopeless the evaluation of a channel’s usefulness for information transmission? It might seem that this is indeed the case. However, this result, as most in information theory relies on asymptotics. But, asymptotics are just a convenient tool to analyse and compare resources that, in the physical world, can only be used a finite number of times. More prosaically, if one asks the question, what is the maximum communications rate over $n$ uses of some channel $N$ with error at most $\epsilon$; then these computability effects seem to fade. The rigorous proof of this statement is worth further investigation.

APPENDIX

In this appendix we adapt the proof of the undecidability of the emptiness problem for our concrete needs and prove the undecidability of the value problem.

A. The emptiness problem

Hirvensalo proved in [22] the undecidability of the emptiness problem for a state size dependent value of $\lambda$. We state this result as Theorem 3.

Theorem 3 (22). Let $k$ be an integer equal or greater than 7 and $(n, m)$ be a duple of integers that is equal or pointwise larger than $(2, 5k − 10)$. Given a PFA with alphabet size $n$ and $m$ states and $1/(5k − 10) = \lambda$ the emptiness of $L_{A>\lambda}$ is undecidable.

Taking Theorem 3 as a starting point, we can amplify the result and obtain undecidability for any $\lambda \in (0, 1)$ computable (e.g. rational) and for stable and resettable PFAs. Now we prove Lemma 1 stated in the main text.

Proof: Given an arbitrary PFA $A = (Q, W, \mathcal{X}, v, F)$ and $p \in (0, 1)$ we are going to construct two PFAs $B_p$ and $C_p$ such that: $L_{A>\lambda}$ is empty $\iff L_{B_p>p\lambda}$ is empty $\iff L_{C_p>p\lambda+1-p}$ is empty.

Let us first construct $B_p = (T, W, \mathcal{Y}, u, F)$. The set of states is $T = \{Q \cup \text{init} \cup \text{sink}\}$. The input alphabet is equal to the original one. For any input symbol $x \in W$ we define the stochastic matrices of $B_p$ as follows:

$$\begin{pmatrix}
X_x & p X_x v \\
0 & 0 \\
0 & 1-p & 1
\end{pmatrix}$$

Note that we have added two rows and columns to track the two new states. Let us parse the action of the automaton as defined by the stochastic matrices. If it is in any of the original states, its behavior remains unchanged. If the automaton is in the sink state no matter what input symbol it reads the PFA remains in the sink state. Finally, if the automaton is in the init state upon reading the input symbol $x$ with probability $1-p$ it will transition to the sink state and with probability $p$ it will transition to whatever the original automaton would have transitioned from the initial distribution. In other words, the new distribution on the states will be given by $(p X_x v, 0, 1-p)$.

The initial distribution of $B_p$ has weight one on the init state, that is: $u = (0, \ldots, 0, 1, 0)$.

The construction of $C_p$ is identical except that we add the sink state to the set of accepting states. We have depicted both constructions in Figure 2.

For any input word $w \in W^*$ we have that $\text{val}(A, w) = p \text{val}(B_p, w) + p \text{val}(C_p, w) + 1 - p$. Hence, $L_{A>\lambda}$ is empty $\iff L_{B_p>p\lambda}$ is empty $\iff L_{C_p>p\lambda+1-p}$ is empty.

B. The value problem

Now we are going to prove the undecidability of the value problem. Our construction for proving the undecidability for
Consider the word $w$. Furthermore, the upper bound is reachable. To verify this, the acceptance value is $y$ from which the automaton can not exit. For any such a word, times the automaton is forced it into the states $\text{sink}, D$. this form the acceptance value is:

$$x > y$$

(Proposition 5 [23])

Lemma 4

The following:

Proof: First, we need to make some observations regarding $D_{x,y}$. If the input letter $b$ is fed two or more consecutive times the automaton is forced it into the states $\text{sink}, q_3$ and $q_6$ from which the automaton can not exit. For any such a word, the acceptance value is $y$. Hence, we concentrate our attention to words of the form $a^{n_1}ba^{n_2}b\ldots ba^{n_k}b$. For any word $w$ of this form the acceptance value is:

$$\text{val}(A, w) = y\ p\left[ q_1 \xrightarrow{w} q_3 \right] + y\ p\left[ q_4 \xrightarrow{w} q_5 \right]$$

(57)

and

$$\text{val}(A, w) = y\ p\left[ q_1 \xrightarrow{w} q_3 \right] + y\ \left( 1 - p\left[ q_4 \xrightarrow{w} q_6 \right] \right)$$

(58)

Furthermore, the upper bound is reachable. To verify this, consider the word $wa^n$, $p\left[ q_1 \xrightarrow{wa^n} q_3 \right]$ does not change and we can make $p\left[ q_4 \xrightarrow{wa^n} q_5 \right]$ approach $1 - p\left[ q_4 \xrightarrow{w} q_6 \right]$ by choosing $n$ large enough. Both $p\left[ q_1 \xrightarrow{w} q_3 \right]$ and $p\left[ q_4 \xrightarrow{w} q_6 \right]$ admit a very compact form:

$$p\left[ q_1 \xrightarrow{w} q_3 \right] = 1 - \prod_{i=1}^{t} (1 - x^{n_i})$$

(59)

$$p\left[ q_4 \xrightarrow{w} q_6 \right] = 1 - \prod_{i=1}^{t} (1 - (1 - x)^{n_i})$$

(60)

$\Rightarrow$ We prove this direction by showing the contrapositive. If $x \leq 1/2$ it implies that $x - 1 - x$ and in consequence

$$1 - \prod_{i=1}^{t} (1 - x^{n_i}) \leq 1 - \prod_{i=1}^{t} (1 - (1 - x)^{n_i}) .$$

(61)

Let $\epsilon > 0$, for any word $w$ such that $p[ q_1 \xrightarrow{w} q_3 ] = 1 - \epsilon$ we have $p[q_4 \xrightarrow{w} q_6] \geq 1 - \epsilon$ and $\text{val}(D_{x,y}, w) \leq y$.

$\Leftarrow$ We are going to prove that for any $\epsilon \in (0, x)$ there exists a word $w$ such that:

$$p\left[ q_4 \xrightarrow{w} q_6 \right] \leq \epsilon$$

(62)

$$p\left[ q_1 \xrightarrow{w} q_3 \right] \geq 1 - \epsilon$$

(63)

Consider the sequence of words $\{w_k\}_{k=2}^{\infty}$ where $w_k = a^{n_2}ba^{n_3}b\ldots ba^{n_k}b$ and the lengths $n_2 \ldots n_k$ are given by

$$n_k = \left[ \log_x \frac{1}{k} + C_\epsilon \right]$$

(64)

and

$$C_\epsilon = \frac{1}{b} \log_x \left( \frac{b - 1}{b} \epsilon \right).$$

(65)

Let $b > 1$ be a number such that $x^b = 1 - x$. The following sequence of inequalities holds:

$$p\left[ q_4 \xrightarrow{w_k} q_6 \right] = (1 - x)^{n_1} + (1 - (1 - x)^{n_2})(1 - x)^{n_3} + \ldots + \prod_{i=2}^{k-1} (1 - (1 - x)^{n_i})(1 - x)^{n_k}$$

(66)

$$\leq \sum_{i=2}^{k} (1 - x)^{n_i}$$

(67)

$$= \sum_{i=2}^{k} x^{bn_i}$$

(68)

$$= \sum_{i=2}^{k} x^{b\lfloor \log_x \frac{1}{k} + C_\epsilon \rfloor}$$

(69)

$$\leq x^{bC_\epsilon} \sum_{i=2}^{k} x^{b\log_x \frac{1}{i}}$$

(70)

$$= x^{bC_\epsilon} \sum_{i=2}^{k} \frac{1}{i^b}$$

(71)

Note that the sum in the right hand side of (71) when $k$ goes to infinity is very similar to the Riemann zeta function evaluated.
at a real argument strictly larger than one. For this arguments it is well known \[34\] that it can be bounded by

\[
\zeta(b) = \sum_{n=1}^{\infty} \frac{1}{n^b} \leq \frac{b}{b-1}.
\]  

(72)

If we apply this bound to \(71\) we obtain

\[
\lim_{k \to \infty} p[q_4 \xrightarrow{w_k} q_6] \leq \lim_{k \to \infty} x^{bc_k} \sum_{i=2}^{k} \frac{1}{i^b}
\]

\[
\leq x^{bc_k} \frac{b}{b-1}
\]

\[
= \epsilon.
\]  

(74)

(75)

Furthermore, \(75\) remains an upper bound for finite \(k\) since we are only dropping positive contributions. Hence, \(62\) is verified for all \(k\). Let us now verify that there exists \(k\) such that the requirement \(63\) also holds. Consider the following sum

\[
\sum_{i=2}^{k} x^{n_i} \geq \sum_{i=2}^{k} x^{\log x \cdot \frac{1}{i} + C_i + 1}
\]

\[
= x^{C_i + 1} \sum_{i=2}^{k} \frac{1}{i}
\]  

(77)

and this sum diverges for any non-zero \(x\) and finite \(C_i\). This implies that \(\lim_{k \to \infty} \prod_{i=2}^{k} (1 - x^{n_i}) = 0\) and that there exists a finite \(k\) such that \(\prod_{i=2}^{k} (1 - x^{n_i}) \leq \epsilon\). Then, \(p[q_1 \xrightarrow{w_k} q_3] \geq 1 - \epsilon\).

Lemma 4 reduces the value problem to \(x\) being larger than one half. Now, we are going to modify \(D_{x,y}\). The main idea is that \(x\) will be replaced by the probability that an automaton \(A\) accepts a word \(w_A\). This is achieved very easily, see Figure 4 once the state of \(D_{A,y}\) reaches \(A\) it continues there until it sees \(c\) which is a symbol outside the input alphabet of \(A\). Then, if \(A\) is in an accepting state it will transition to the same state while it will transition to a different state if it is not in an accepting state. We indicate the transitions from an accepting state by \(\rightarrow\) and the transitions from a non-accepting state by \(\Rightarrow\). Let \(w_A\) be an arbitrary input word into \(A\) then:

\[
p[q_1 \xrightarrow{aw_{A,c}} q_1] = val(A, w_A)
\]

(78)

\[
p[q_4 \xrightarrow{aw_{A,c}} q_5] = val(A, w_A)
\]  

(79)

In the following we prove Theorem 2.

**Proof:** We will prove the statement by reducing the value problem to the emptiness problem. More precisely, we will prove that \(\text{val}(D_{A,y}, w) < 2y\) if and only if \(L_{A>1/2}\) is empty.

\(\Rightarrow\): Assume that \(L_{A>1/2}\) is not empty. Then there exists some \(w_A\) such that \(\text{val}(A, w_A) > 1/2\). Hence, we can construct the sequence \(w_k = (aw_{A,c})^{n_1} \ldots (aw_{A,c})^{n_k}\) with the lengths \(n_2 \ldots n_k\) given by \([64]\). Following the proof of Lemma 4 we have that for \(\epsilon > 0\) there exists \(k\) such that \(w_k\) verifies conditions \([62]\) and \([63]\).

\(\Leftarrow\): We can restrict our attention to words of the form \((aw_{A,c})^{n_1} b \ldots b (aw_{A,c})^{n_k} b\). Furthermore for any word \(w\) we have that \(\text{val}(A, w) \leq 1 - \text{val}(A, w_A)\) and in consequence

\[
1 - \prod_{i=1}^{k} (1 - \text{val}(A, w_A^{n_i})) \leq 1 - \prod_{i=1}^{k} (1 - (1 - \text{val}(A, w_A^{n_i})))
\]

(80)

Let \(\epsilon > 0\) \([80]\) implies that for any word such that \(p[q_1 \xrightarrow{w} q_4] = 1 - \epsilon\) we have that \(p[q_4 \xrightarrow{w} q_5] \geq 1 - \epsilon\) and \(\text{val}(D_{A,y}, w) \leq y\).

**C. Resettable and stable channels**

We prove Lemma 2 stated in the main text.

**Proof:** We are going to reduce the problem for stable and resettable PFAs to that of arbitrary PFA. For this we will start with pairs \(A\) and \(\lambda_0\) for which the emptiness problem is undecidable. Now, for the emptiness property with \(\lambda\) we choose \(\tilde{A} = B_{\lambda / \lambda_0}\) if \(\lambda \leq \lambda_0\) and \(\tilde{A} = C_{(\lambda - 1) / (\lambda_0 - 1)}\) if \(\lambda > \lambda_0\). For the value problem we choose \(\tilde{A} = D_{A,\lambda / 2}\). Using Lemma 1 and Theorem 2 it is then enough to show that for \(\tilde{A}\):

1) \(L_{\tilde{A} > \lambda}\) is empty if and only if \(L_{\gamma(\tilde{A}) > \lambda}\) is empty.

2) \(\text{val}_{\tilde{A}} \lambda\) if and only if \(\text{val}_{\gamma(\tilde{A})} \lambda\)

\(\Leftarrow\) This direction of both statements is trivial since the set of input words of \(L_{\gamma(\tilde{A})}\) is a strict superset of the set of input words of \(L_{\tilde{A}}\).
Let us divide the input words into two sets: $W_1$, the words that either end with the symbol rt or consist of a string of id and $W_2$ which is the complementary set, that is, words that have at least one symbol different than id and do not end with the rt symbol. The acceptance probability of any $w \in W_1$ is simply the acceptance probability of a distribution with unit probability on the initial symbol. Since for $B_{\gamma}(\lambda_0)$, $C(\lambda_1)/(\lambda_0-\lambda_1)$ and $D_{\gamma}(\lambda_0)$ the acceptance and initial symbols are disjoint, the value of $w$ is zero. That means that no word from $W_1$ can be in the set $\{w: \text{val}(\gamma(A), w) \geq \lambda\}$ for any value of $\lambda \in (0, 1]$.

First, consider any word $w \in W_2$ that contains at least one identity symbol, it can be written as $w_1dw_2$ where $w_1$ and $w_2$ are two sequences of input symbols and at least one of both is non-empty. We have that $\text{val}(\gamma(A), w) = \text{val}(\gamma(A), w_1w_2)$ and by applying this argument to all the identity symbols in the word we find a new word $w'$ with no identity symbols such that $\text{val}(\gamma(A), w) = \text{val}(\gamma(A), w')$. Hence we can restrict our attention to words with no identity symbol.

Second, we consider any word $w \in W_2$ that contains at least one reset symbol, it can be written as $w_1rtw_2$ where at least $w_2$ is non-empty. We have that $\text{val}(\gamma(A), w) = \text{val}(\gamma(A), w_1w_2)$, again we can apply this argument to all the reset symbols in the word and find a word $w'$ with no reset or identity symbols such that $\text{val}(\gamma(A), w) = \text{val}(\gamma(A), w') = \text{val}(A, w')$.

ACKNOWLEDGEMENTS

The authors are grateful to Toby Cubitt and Michael Wolf for stimulating discussions. We thank Kenneth Goodenough for comments on an earlier version of the manuscript and for pointing us to the upper bound on the Riemann zeta function.

REFERENCES