Polar codes with a stepped boundary

Ilya Dumer
February 15, 2017

Abstract: We consider explicit polar constructions of block-length $n \to \infty$ for the two extreme cases of code rates $R \to 1$ and $R \to 0$. For code rates $R \to 1$, we design codes with complexity order of $n \log n$ in code construction, encoding, and decoding. These codes achieve the vanishing output bit error rates on the binary symmetric channels with any transition error probability $p \to 0$ and perform this task with a substantially smaller redundancy $(1 - R)n$ than do other known high-rate codes, such as BCH codes or Reed-Muller (RM). We then extend our design to the low-rate codes that achieve the vanishing output error rates with the same complexity order of $n \log n$ and an asymptotically optimal code rate $R \to 0$ for the case of $p \to 1/2$.

Keywords: Polar codes; Reed-Muller codes; Boolean polynomials; successive cancellation decoding.

I. INTRODUCTION

Below we consider the Plotkin recursive construction $u, u + v$ that repeatedly combines shorter codes to construct and decode the longer ones. RM codes $R(r, m)$ represent one Plotkin-type construction [1] of length $n = 2^m$ and dimension $k(r, m) = \sum_0^r \binom{m}{r}$ with parameters $0 \leq r \leq m$. Polar codes [3] introduce another recursive design. Both codes originate from the same full-space code $R(m, m)$ and filter it in two different ways. Namely, a code $R(r, m)$ maximizes the code rate among all codes that have the same distance $2^{m-r}$ and are generated by the $m$-variate Boolean monomials. Polar codes use a more intricate optimization. First, the successive-cancellation decoding (SCD) of [2]-[6] performs step-by-step retrieval of information bits of code $R(m, m)$. Analysis of SCD [6] shows that it yields both high and low-fidelity information bits for RM codes. Therefore, removing low-fidelity bits (by setting them as zeros) gives the better-performing subcodes of RM codes. For relatively short lengths of 512 or less, this was done in [5], [6]. In particular, it turns out that these subcodes achieve a nearly optimal (ML) performance on these lengths if SCD is combined with list decoding. For long codes with $m \to \infty$, the major breakthrough achieved in [3] shows that the subcodes of $R(m, m)$ that keep $Rn$ most reliable bits are capacity achieving (CA) codes under SCD for any binary symmetric memoryless channel $U$ and any code rate $R \in (0, 1)$. These polar codes also achieve a polynomial complexity of construction. Namely, for a channel $U$ with capacity $C$, polar codes of code rate $R > C - \epsilon$ have complexity $\text{poly}(a^{-m})$ for any $\epsilon > 0$, where $a = a(U)$ and $\mu$ are some constants.

Below, we extend the above results for the special cases of $R \to 1$ and $R \to 0$. In both cases, we consider code families that achieve a vanishing output bit error rate on a binary symmetric channel $BSC(p)$ with a transition error probability $p$ and capacity $C = 1 - h(p)$, where $h(p)$ is a binary entropy. We say that a family of codes with $n \to \infty$ and $R \to 1$ is strongly optimal if the fraction $\rho = 1 - R$ of redundant (parity-check) bits has the smallest possible order

$$1 - R \sim p \log_2 \frac{C}{p} + O(p^2)$$

A family of long codes is called weakly optimal if probability $p \to 0$ and redundancy $\rho$ have a similar decline rate

$$\log_2(1 - R) \sim \log_2 h(p) \sim \log_2 p$$

Our main result is as follows.

Theorem 1. For any $p \to 0$, there exist weakly optimal codes of length $n \to \infty$ that have a relative redundancy

$$\rho \leq p \left(\log_2 \frac{1}{p}\right)^{\log_2 \log_2 \frac{1}{p}}$$

and achieve a vanishing error probability on a binary symmetric channel $BSC(p)$. These codes can be constructed, encoded, and decoded with complexity of order $n \ln n$.

Similarly, long codes of rate $R \to 0$ are called strongly optimal if they achieve a vanishing output error rate on a $BSC(p)$ with $p \to 1/2$ and have the maximum possible order of code rate $R \sim 1 - h(p) \sim (1 - 2p)^2 / \ln 4$. We extend Theorem 1 and design strongly optimal codes of rate $R \to 0$ and complexity $n \ln n$.

For a wide range of error probabilities $p$, codes of Theorem 1 outperform known codes of code rate $R \to 1$. For example, long primitive BCH codes require redundancy $p \log_2 n$ to achieve a vanishing output error rate under the bounded-distance decoding on a BSC if $p = o(\log_2 n)$ [1]. However, $R \to 0$ if $p \log_2 n \to \infty$. The recent breakthrough of [12] also shows that high-rate RM codes $R(m - 2r - 1, m)$ can correct the fraction of errors $p \sim \binom{m}{r} / 2^m$ with polynomial complexity and low redundancy $\rho \sim \binom{m}{2r+1} / 2^m$ if $r = \sqrt{\ln m} / \ln m$. This algorithm is still limited to the rapidly vanishing probabilities $p$ unlike any $p \to 0$ in Theorem 1. Note, however, that Theorem 1 achieves no improvements over BCH codes if probability $p$ has an exponentially declining order $p \leq 2^{-m^n}$ for any $n > 0$, nor does it give strongly optimal codes for $R \to 1$.

Sections [II] and [III] provide some background and address the common properties of RM and polar codes. Sections [IV] and [V] introduce polarized design with a single boundary. We first design the weakly optimal codes of rates $R \to 1$ and then extend them to the strongly optimal codes of rate $R \to 0$. 

I. Dumer is with the College of Engineering, University of California, Riverside, CA 92521, USA; email: dumer@ee.ucr.edu
II. RECURSIVE DESIGN OF RM AND POLAR CODES

Consider boolean polynomials $f(x)$ of degree $r$ or less in $m$ binary variables $x_1, \ldots, x_m$, where $r \leq m$. Vectors $x = (x_1, \ldots, x_m)$ will mark the positions of our code. Each map $f(x) : \mathbb{F}_2^m \rightarrow \mathbb{F}_2$ generates a codeword $c = c(f)$ of code $\mathcal{R}(r, m)$. We also use short notation $x_{i|j} = (x_i, \ldots, x_j)$ for $i \leq j$. Consider a recursive decomposition

$$f(x) = f_0(x_{|2}) + x_1 f_1(x_{|2}) = \sum_{i_1, \ldots, i_{\ell}} x_1^{i_1} \cdots x_\ell^{i_\ell} f_{i_1, \ldots, i_{\ell}}(x_{\ell+1|m})$$

The first step decomposes polynomial $f(x)$ into monomials $f_0$ and $f_1$ of degrees $d_0 \leq \min\{r, m-1\}$ and $d_1 \leq r-1$. Then the codewords $c_0 = c(f_0)$ and $c_1 = c(f_1)$ belong to the codes $\mathcal{R}(r, m-1)$ and $\mathcal{R}(r-1, m-1)$ and form the codeword $c = c_0 + c_1$ of code $\mathcal{R}(r, m)$. Similarly, any subsequent step $\ell$ decomposes each polynomial with respect to $x_\ell$ as follows

$$f_{i_1, \ldots, i_{\ell-1}}(x_{\ell|1}) = \sum_{i_\ell = 0, 1} f_{i_1, \ldots, i_{\ell-1}}(x_{\ell+1|m}) x_\ell^{i_\ell}$$

We then say that the $\ell$-level binary paths $\xi_1 = i_1, \ldots, i_\ell$ decompose the original polynomial $f(x)$ into sums of monomials $x_1^{i_1} \cdots x_\ell^{i_\ell} f_{i_1, \ldots, i_{\ell}}(x_{\ell+1|m})$. Finally, full paths $\xi = i_1, \ldots, i_m$ of step $m$ define monomials $x^\xi = x_1^{i_1} \cdots x_m^{i_m}$ with coefficients $f_\xi = f_{i_1, \ldots, i_m} = 0, 1$. Note that each monomial $x^\xi$ gives a codeword $c(x^\xi)$ of weight $2^{m-\ell}w(\xi)$, where $w(\xi)$ is the Hamming weight of the string $\xi$. RM codes $\mathcal{R}(r, m)$ include only $k(r, m)$ paths of weight $w(\xi) \leq r$.

In Fig. 1, we use this representation for the full code $\mathcal{R}(4, 4)$. Each decomposition step $\ell = 1, 2, 4$ is marked by the splitting monomial $x_\ell^{i_\ell}$. For example, path $\xi = 0110$ gives the coefficient $f_{0110}$ associated with the monomial $x^\xi = x_2 x_3$.

Fig. 2 depicts code $\mathcal{R}(2, 5)$. Here we only include all paths $\xi$ of weight $w(\xi) \leq 2$. Note that any two paths $\xi_1, \xi_2$ entering some node have the same weight $w$ and generate the same code $\mathcal{R}(r - w, m - \ell)$ on their extensions. For example, path $\xi = 01100$ proceeds from $\mathcal{R}(2, 5)$ to the single bit $\mathcal{R}(0, 0)$ via codes $\mathcal{R}(2, 4), \mathcal{R}(1, 3), \mathcal{R}(0, 2)$, and $\mathcal{R}(0, 1)$.

This design can be reformulated using a $2 \times 2$ matrix

$$G = \begin{bmatrix} 1 & 1 \\ 0 & 1 \end{bmatrix}$$

Then code $\mathcal{R}(m, m)$ is generated by the Kronecker product $G(m, m) = G^\otimes m$. Each row of $G^\otimes m$ is the map of the monomial $x^\xi$ for some path $\xi$. Similarly, matrix $G(r, m)$ is the map of all monomials $x^\xi$ with paths $\xi = i_1, \ldots, i_m$ of weight $w(\xi) \leq r$.

Now consider a single path $\xi$ that ends with an information bit $f_{i_1, \ldots, i_m} = 1$. Encoding proceeds in the reverse order $\ell = m, \ldots, 1$. We begin with a single bit codeword $c(\xi_{m+1|m}) = 1$.

In each step $\ell$, we use recursion and obtain the codeword

$$c(\xi_{\ell|m}) = \begin{cases} c(\xi_{\ell+1|m}) & \text{if } i_\ell = 0 \\ 0, c(\xi_{\ell+1|m}) & \text{if } i_\ell = 1 \end{cases}$$

of length $2^{m-\ell+1}$. Thus, any path $\xi$ is encoded in the vector $c = c(\xi)$ of length $n$. Also, $c(\xi) = 0$ if $f_{i_1, \ldots, i_m} = 0$.

Now consider a subset of $N$ paths $T$. Then we encode $N$ information bits via their paths and obtain codewords $c(T) = \sum_{\xi \in T} c(\xi)$. These codewords form a linear code $C(m, T)$. Here at any level $\ell$, encoding adds two codewords of level $\ell + 1$ to any node $\xi_{\ell|m}$. Thus, encoding $\ell$ performs $2^{m-\ell}$ operations on each of $2^{\ell}$ nodes $\xi_{\ell|m}$ and has the overall complexity of $n \log_2 n$ over all levels $\ell$.

Lemma 2. Code $C(m, T)$ has length $2^n$, dimension $|T|$ and distance $2^{m-r}$, where $r = \max\{w(\xi), \xi \in T\}$ is the weight of the heaviest path in $T$. Code $\mathcal{R}(r, m)$ has the maximum code rate $R$ among all codes $C(m, T)$ of the distance $2^{m-r}$.

Proof. Let weight $r$ be achieved on some path $\psi \in T$. Then code $C(m, T)$ is generated by monomials $x^\xi$ of degree $r$ or less. Thus, $C(m, T) \subseteq \mathcal{R}(r, m)$. The monomial $x^\psi$ has degree $r$ and gives the minimum weight $2^{m-r}$.

III. RECURSIVE DECODING ALGORITHMS

Below, we use a map $x \rightarrow (-1)^x$ for any $x = 0, 1$ and consider a discrete memoryless channel (DMC) $W$ with inputs $\pm 1$. Vector $ab$ will denote the component-wise product of vectors $a, b$ and $c = (u, uv)$ will denote the codewords $c$ of a code $\mathcal{R}(r, m)$ with symbols $\pm 1$. In particular, $1^n$ now represents a former all-zero codeword. For any codeword $c$, let $y_0, y_1$ be the two output halves corrupted by noise. We use double index $i, j$ for any position $j = 1, \ldots, n/2$ in a half $i = 0, 1$. Define the posterior probability (PP) $g_{i,j} = \Pr[c_{i,j} = 1 | y_{i,j}]$ that $1$ is sent in position $i, j$. We will often replace $g_{i,j}$ with two related quantities, which we call “the offsets” $q_{i,j}$ and the likelihoods $h_{i,j}$:

$$g_{i,j} = 2q_{i,j} - 1, \quad h_{i,j} = q_{i,j}/(1 - q_{i,j})$$

Thus, we will use vectors $q = (q_{i,j})$, $g = (g_{i,j})$ and $h = (h_{i,j})$. For example, let $W$ be a binary symmetric channel.
BSC(p), where \( p = (1 - e)/2 \). Then any output \( y = \pm 1 \) gives quantities \( g(y) = ey \) and \( h(y) = (1 + ey)/(1 - ey) \).

The following recursive algorithm \( \Psi_m^p(q) \) of [2], [5] performs SCD of information bits in codes \( R(r, m) \) or their subcodes \( C(m, T) \). Here we relegate decoding of vector \( q \) to two vectors \( q^{(1)} \) and \( q^{(0)} \) of length \( n/2 \). Vector \( q^{(1)} \) consists of PP \( g_j^{(1)} = Pr\{v_j = 1 | g_{0,j}, g_{1,j}\} \) of symbols \( v_j \) in construction \((u, uv)\). Simple recalculations [2] show that the offsets \( g_j^{(1)} \) of symbols \( v_j \) can be expressed as the products of two offsets \( g_{0,j}g_{1,j} \). Thus, we obtain vectors \( g^{(1)} \) and \( q^{(1)} \) with symbols

\[
G_j^{(1)} = g_{0,j}g_{1,j}, \quad Q_j^{(1)} = (1 + G_j^{(1)})/2. \tag{6}
\]

We may now apply some decoding algorithm \( \Psi_{m-1}^{p}(q) \) to the vector \( q^{(1)} \) and obtain a vector \( \bar{v} \in R(r-1, m-1) \) of length \( n/2 \). Now we have two corrupted versions \( y_0 \) and \( y_1 \bar{v} \) of vector \( u \). We can then derive PP \( q_j^{(0)} = Pr\{u_j = 1 | g_{0,j}, g_{1,j}, \bar{v}_j\} \) of symbols \( u_j \) in \((u, uv)\) construction. Indeed, any symbol \( u_j \) has likelihoods \( h_{0,j} \) and \( (h_{1,j})^{\bar{v}_j} \) in the left and right halves, respectively. Then we combine the two likelihoods into their product:

\[
h_j^{(0)} = h_{0,j}(1 + h_{1,j})^{\bar{v}_j}, \quad Q_j^{(0)} = h_j^{(0)}/(1 + h_j^{(0)}) \tag{7}
\]

Then we can apply some decoding \( \Psi_{m-1}^{p}(q) \) to vector \( q^{(0)} \) and obtain \( \bar{u} \in R(r, m-1) \).

Decomposition \( 6, 7 \) forms level \( \ell = 1 \) of SCD, which can also be continued for vectors \( q^{(1)} \) and \( q^{(0)} \) on the codes \( R(r-1, m-1) \) and \( R(r, m-1) \). Then levels \( \ell = 2, \ldots, m \) are processed similarly, moving along the paths of Fig. 1 or Fig. 2. Any incomplete path \( \xi_{1|\ell} \) begins with its \( v \)-extension \((\xi_{1|\ell}, 1) \). Upon decoding, this path delivers its output \( \bar{v} \) to the \( u \)-path \((\xi_{1|\ell}, 0) \). Thus, all paths are ordered lexicographically. Finally, the last step gives the likelihood of \( g_{\ell} = Pr\{f_{\ell} = 0 | y_0, y_1\} \) of one information bit \( f_{\ell} \) on the path \( \xi \). We then choose the more reliable bit \( f_{\ell} \). It is easy to verify [2] that \( m \) decomposition steps give complexity \( 2n \log_2 n \).

Any subcode \( C(m, T) \) is decoded similarly and assumes that all paths \( \xi \notin T \) are frozen and give information bits \( f_{\ell} \equiv 0 \). Let all \( N \) paths in \( T \) be ordered lexicographically as \( \xi^{(1)}, \ldots, \xi^{(N)} \). Then we have

\[
\text{Algorithm } \Psi(m, T) \text{ for code } C(m, T).
\]

Given: a vector \( q = (g_{0,j}, g_{1,j}) \) of PP. Take \( s = 1, \ldots, N \) and \( \ell = 1, \ldots, m \).

For path \( \xi^{(s)} = i^{(s)}_1, \ldots, i^{(s)}_m \) in step \( \ell \) do:

Apply recalculations \( 6 \) if \( i^{(s)}_\ell = 1 \)

Apply recalculations \( 7 \) if \( i^{(s)}_\ell = 0 \).

Output the bit \( f_{\xi^{(s)}} \) for \( \ell = m \).

IV. PATH ORDERING IN SC DECODING

Let a binary code \( C(m, T) \) be used over a symmetric DMC \( W \). We now consider a code \( C_\xi \) defined by a single path \( \xi = (i_1, \ldots, i_m) \) and estimate its decoding error probability \( P_\xi \). Let a codeword \( 1^n \) be transmitted over this path. We now may assume that other paths give outputs \( \bar{v}_j = 1 \) in recursive recalculations [5-7]. Then we re-arrange [5-7] as follows

\[
g_j^{(1)} = g_{0,j}g_{1,j}, \quad g_j^{(0)} = (g_{0,j} + g_{1,j})/(1 + g_{0,j}g_{1,j}) \tag{8}
\]

\[
h_j^{(0)} = h_{0,j}h_{1,j}, \quad h_j^{(1)} = (1 + h_{0,j}h_{1,j})/(h_{0,j} + h_{1,j}) \tag{9}
\]

From now on, we may consider recalculations [8] and [9] as the sequences of channel transformations applied to the original random variables \((r) g_{0,j}, h_{1,j} \). In the end, we obtain a new memoryless channel \( W_\xi : X \rightarrow Y_\xi \) that outputs a single \( r v \ h_\xi \) after \( m \) steps. For any parameter \( \lambda > 0 \), we also consider \( r v \ h_\xi \) and its expectation \( E h_\xi - \lambda \). Then the Chernoff upper bound gives

\[
P_\xi \equiv Pr\{h_\xi < 1\} \leq \min_{\lambda > 0} E h_\xi - \lambda \text{ min } E e^{-\lambda \ln h_\xi} \]

Note that the quantity \( E h_\xi - \lambda \) is identical to the Bhattacharyya parameter

\[
Z(W) = \sum_{y \in Y} \sqrt{W(y|0)} \sqrt{W(y|1)}
\]

defined for a DMC channel \( W_\xi : X \rightarrow Y_\xi \). For example, BSC(p) with \( p = (1 - g)/2 \) gives

\[
Z(W) = E h_\xi - \lambda \leq 2 \left( \frac{1-g}{2} \right)^{1/2} (1-g)^{1/2} = \sqrt{1-g^2}
\]

In a more general setting [7], we decompose a binary symmetric DMC \( W_\xi \) into some number \( k \) of binary symmetric channels \( BSC_{\theta_i}(p_i) \) that have transition error probabilities \( p_i = (1 - g_i)/2 \) and occur with some probability distribution \((\theta_i)\), where \( \sum_i \theta_i = 1 \).

\[
Z(W_\xi) = \sum_i \theta_i \sqrt{1-g_i^2} \tag{10}
\]

Below we use the upper bound \( P_{\xi} \leq Z(W_\xi) \) employed by Arikan in [4]. It is also proved in [3] that a one step recursion \((W, W) \rightarrow (W^{(1)}, W^{(0)})\) of [9] gives parameters \( Z(W^{(1)}) \) and \( Z(W^{(0)}) \) such that

\[
1 - Z(W^{(1)}) \geq (1 - Z(W))^2, \quad Z(W^{(0)}) = Z^2(W) \tag{11}
\]

Now consider a compound channel \( W_\xi \) as a set of \( BSC_{\theta_i}(p_i) \).

Then we can define the expectation of the offsets \( g_i > 0 \):

\[
G(W_\xi) = \sum_i \theta_i g_i
\]

Note that \( \sqrt{1-g^2} \) is a concave function. Also, \( \sqrt{1-g^2} \geq 1-g \) for any \( g \in [0, 1] \). Thus, (10) yields two inequalities

\[
1 - G(W_\xi) \leq Z(W_\xi) \leq \sqrt{1 - |G(W_\xi)|^2} \tag{12}
\]

Given a one step recursion \((W, W) \rightarrow (W^{(1)}, W^{(0)})\), we can also take two independent identically distributed rv \( g_{0,j}, g_{1,j} \) in [8] and find the expectation of their product \( g_j^{(1)} \) for the channel \( W^{(1)} \). Then we have two equalities

\[
G(W^{(1)}) = G^2(W), \tag{13}
\]

\[
Z(W^{(0)}) = Z^2(W) \tag{14}
\]

Below we replace notation \( Z(W_\xi) \) and \( G(W_\xi) \) with \( Z(\xi) \) and \( G(\xi) \). Given a path \( \xi = (i_1, \ldots, i_m) \), we say that a path \( \eta =
Step 3: \( i_s = 1 \Rightarrow j_s = 0, \) \( (i_s = 1, i_{s+1} = 0) \Rightarrow (j_s = 0, j_{s+1} = 1) \) (15)

Let \( h(\xi) \) and \( h(\eta) \) be the outputs of paths \( \xi \) and \( \eta \) obtained by recalculations (9). The following Lemma 3 uses a partial order for the paths \( \xi \) and \( \eta \) with respect to the quantities \( \mathbb{E} h^{-\lambda}(\xi) \). A similar lemma was used in (10) for a slightly different set of recalculations, which approximate recurrences (9). In (8) and (9), this lemma is proved for the Bhattacharyya parameter \( Z(\mathcal{W}_x) \) with exact recalculations (9). In Appendix, we also post a proof of Lemma 3 for the arbitrary moments \( \mathbb{E} h^{-\lambda}(\xi) \).

**Lemma 3.** Recalculations (9) on some path \( \xi \) and its descendant \( \eta \) give the outputs \( h(\xi) \) and \( h(\eta) \) that satisfy inequalities

\[
\mathbb{E} h^{-\lambda}(\xi) \geq \mathbb{E} h^{-\lambda}(\eta), \quad \lambda \in [0, 1], \tag{16}
\]

\[
\mathbb{E} h^{-\lambda}(\xi) \leq \mathbb{E} h^{-\lambda}(\eta), \quad \lambda \in [1, \infty). \tag{17}
\]

**Corollary.** Any path \( \xi \) and its descendant \( \eta \) satisfy inequalities

\[
P(\eta) \leq Z(\eta) \leq Z(\xi). \tag{20}
\]

Below, we say that a path \( \xi \) forms a **boundary** for all descendant paths \( \eta \) that satisfy replacements (15).

**V. HIGH-RATE CODES WITH A STEPPED BOUNDARY**

Below, \( \log x \equiv \log_2 x \). For \( i = 1, \ldots, s \), consider a set of 2s non-negative ordered integers \( \mathcal{L} = \{r_i, \ell_i\} \) such that \( r_i + \ell_i = m_i \) and \( \sum_{i=1}^{s} m_i = m \). We say that a path

\[
\xi(\mathcal{L}) = \xi^{(1)}, \ldots, \xi^{(s)} = 1^i 0^{\ell_i}, \ldots, 1^r 0^{\ell_s} \tag{18}
\]

of length \( m \) bounds a path \( \eta(\mathcal{L}) = \eta^{(1)}, \ldots, \eta^{(s)} \) if each section \( \eta^{(i)} \) of length \( m_i \) has weight

\[
w(\eta^{(i)}) \leq r_i, \quad i = 1, \ldots, s. \tag{19}
\]

Thus, each section \( \eta^{(i)} \) is located to the right of \( \xi^{(i)} \) as seen in Fig. 3 for a path \( \xi(\mathcal{L}) = 1^{i_1} 0^{\ell_1} 1^{i_2} 0^{\ell_2} 1^{i_3} 0^{\ell_3} \). Clearly, any path \( \eta(\mathcal{L}) \) satisfies (15).

**Proof.** Each segment \( \xi^{(i)} \) is a boundary for the subpaths \( \eta^{(i)}: w(\eta^{(i)}) \leq r_i \). These single-step subpaths span the code \( \mathcal{R}(r_i, m_i) \), which is generated by monomials of degree \( r_i \) or less taken over variables \( x_{M_i+1}, \ldots, x_{M_i+m_i} \), where \( M_i = \sum_{j=1}^{i-1} m_j, M_1 = 0 \). Thus, \( \mathcal{R}(\mathcal{L}) \) is the direct product of codes \( \mathcal{R}(r_i, m_i) \) and has rate \( R_L \). Each row \( \eta \) of its generator matrix is a map \( x^0: \mathbb{F}_2^m \rightarrow \mathbb{F}_2 \) of the monomial \( x^0 \) defined by a path \( \eta \). Thus, conditions (19) require \( m \) operations to verify that \( \eta \in \eta(\mathcal{L}) \) for any row \( \eta \). For \( n \)-row verification, the complexity is bounded by the order \( n \log n \).

Consider a sequence of channels \( \text{BSC}(p) \) with \( p \rightarrow 0 \). Let \( s = o(\log m) \) be some integer. We take a set of 2s numbers

\[
\mathcal{L} = \{r_i = 2^{i-1} \log (1/p) - c_i, \quad \ell_i = 2^{i-1} \log \log (1/p) \} \tag{21}
\]

where \( c_1 = 6, c_2 = 3 \) and \( c_i = 0 \) for \( i \geq 3 \). We also assume that \( \mathcal{L} \) is a set of integers. Then the path \( \xi(\mathcal{L}) \) has the length

\[
m = \sum_{i=1}^{s} m_i = (2^s - 1)(\log (1/p) + \log \log (1/p)) - 9 \tag{22}
\]

An equivalent setting arises if \( p \sim m^{2m/(2^m-1)} \) for \( m \rightarrow \infty \). Note that the case \( s = 1 \) gives a single code \( \mathcal{R}(r, m) \) with \( m-r \sim \log m \) and \( p \sim m^{2m-m} \). We first estimate the redundancy \( \rho_L \) of a code \( \mathcal{R}(\mathcal{L}) \) with boundary (21).

**Lemma 5.** Codes \( \mathcal{R}(\mathcal{L}) \) with a boundary \( \mathcal{L} \) of (21) satisfy the redundancy bound (2) for \( p \rightarrow 0 \).

**Proof.** Let \( \rho_i = 1 - R_i \) denote the redundancy of code \( \mathcal{R}(r_i, m_i) \) used in step \( i \) of design (20). Then

\[
\rho_L = 1 - R_L = 1 - \prod_{i=1}^{s} (1 - \rho_i) \leq \sum_{i=1}^{s} \rho_i \tag{23}
\]

Let \( \tau = \log (1/p) \). First, note that all codes \( \mathcal{R}(r_i, m_i) \) have \( \rho_i \rightarrow 0 \) as \( p \rightarrow 0 \). Indeed, \( \ell_i/m_i \leq (\log \tau)/\tau \rightarrow 0 \) and

\[
\rho_i \leq 2^{-m_i} (\ell_i/m_i) \leq 2^{-m_i[1-h(\ell_i/m_i)]} \tag{24}
\]

Second, note that each segment \( \xi^{(i)} \) of boundary (21) has the length \( m_i \geq 2m_i-1 \). Then \( \rho_i \sim o(\rho_i-1) \) and \( \rho_L \sim \rho_i \). Finally, we use the bounds

\[
h(\ell_i/m_i) = (\ell_i/m_i) \log (c_{\ell_i}/m_i) + O(\ell_i^2/m_i^2) \tag{25}
\]

\[
\rho_i \sim (64p/\tau)(\varepsilon \tau/\log \tau)^{\log \tau} < p(\varepsilon \tau/\log \tau) \tag{26}
\]

Thus, \( \rho_L \) and \( \rho_L \) satisfy asymptotic bound (2).

We now can prove Theorem 1 for construction (21). Here we use the same approach that employed the boundary paths in (2) and (10); however, we extend this approach to a multi-step boundary (21) instead of the single--step and double-step boundaries used before. We proceed as follows. Consider any high-quality channel \( W \), such as \( \text{BSC}(p) \) with \( p \rightarrow 0 \), and its two descendant channels \( W^{(1)} \) and \( W^{(0)} \). Note that the degrading channel \( W^{(1)} \) and the upgrading channel \( W^{(0)} \) exhibit a vastly different behavior. In particular, let the original parameter \( Z(W) \sim \delta \) be close to 0 and the complementary parameter \( G(W) \in [1-\delta,(1-\delta^2)^{1/2}] \) be close to 1. Then the channel \( W^{(0)} \) undergoes a sharp improvement over \( W \) and yields an exponentially declining parameter \( Z(W^{(0)}) \sim \delta^2 \), according to (14). By contrast, the channel \( W^{(1)} \) experiences a relatively small degradation and yields \( G(W^{(1)}) \in [1-2\delta, 1-

---

**Fig. 3.** Polar code with 3 steps 1^00^1

**Lemma 4.** Paths \( \eta(\mathcal{L}) \) of (19) generate the direct product

\[
\mathcal{R}(\mathcal{L}) = \otimes_{i=1}^{s} \mathcal{R}(r_i, m_i) \tag{20}
\]

of \( s \) RM codes \( \mathcal{R}(r_i, m_i) \) of rates \( R_i = k(r_i, m_i)/2^m \). Code \( \mathcal{R}(\mathcal{L}) \) has length \( 2^m \), code rate \( R_L = \Pi_{i=1}^{s} R_i \), and construction complexity of order \( n \log n \).
Then the segment performance in each step of the boundary (21). It is this
the same sign. Then we write Eq. (22). Consider two functions $f = f(n)$ and $r = r(n)$ that have
the same sign. Then we write $f \lesssim r$ or $f \gtrsim r$ if the asymptotic ratio $\lambda = \lim_{n \to \infty} f(r) = \lambda \in (0,1)$ or $\lambda \geq 1$, respectively. We also write $f \asymp r$ if $f \gtrsim r$ for some $c > 1$. Finally, consider inequalities

$$-x - x^2 \leq \ln(1 - x) < -x, \quad x \in (0, 1/2)$$

which are tight as $x \to 0$ and $x \to 1$, respectively. Using these inequalities, we can rewrite (12) as

$$\log Z(\xi) < \frac{1}{2} \log [-\ln G(\xi)]$$

and

$$\ln G(\xi) > -Z(\xi) - Z^2(\xi)$$

Below, we extensively use a recursion that employs inequalities (24) and (25). We will also see that $Z(\xi) \to 0$ and $G(\xi) \to 1$ for the selected path $\xi(L)$ of (21). In this case, we can also replace (24) and (25) with simpler inequalities

$$\log Z(\xi) < \frac{1}{2} \log [-\ln G(\xi)]$$

and

$$\ln G(\xi) > -Z(\xi) - Z^2(\xi)$$

Lemma 6. Codes $\mathcal{R}(L)$ with a boundary (27) achieve an output bit error rate $P_o \to 0$ for each path $\eta(L)$ under SCD on a BSC(p) with $p \to 0$.

Proof. Given the boundary $\xi(L)$, we will estimate the Bhattacharyya parameters

$$Z^{(i)} = Z \left[1^{r_i}0^{\xi_i}1^{r_i/2} \right], \quad Z^{(i)} = Z \left[1^{r_i}0^{\xi_i}1^{r_i/2} \right]$$

obtained in processing of each step $i$. We also use similar notation $G^{(i)}$ and $G^{(i)}$ for the offsets obtained in step $i$. The original channel $BSC(p)$ gives parameter $G = 1 - 2p$, where $p \to 0$. For the first segment $1^{r_1}$, equality (13) and the upper bound (12) give:

$$G^{(1)} = (1 - 2p)^{1/64p} \sim e^{-1/32}$$

$$Z^{(1)} \lesssim \left(1 - e^{-1/16}\right)^{1/2} < 2^{-2}$$

For the next segment $0^{\xi_1}$, equality (14) gives

$$Z^{(1)} = \left[Z^{(1)} \right]^{2\xi_1} < 2^{-2 \log r_{\xi_i}} = p^2$$

Then $G^{(1)} \geq 1 - Z^{(1)}$, according to (12), and we proceed with the segment $1^{r_{\xi_i}}0^{\xi_2}$ using (13):

$$G^{(2)} \geq (1 - p^2)^{r_{\xi_i}/8} \sim e^{-1/8}$$

$$Z^{(2)} \lesssim \left(1 - e^{-1/4}\right)^{1/2} < 1/2$$

$$Z^{(2)} = \left[Z^{(2)} \right]^{2\xi_2} < 2^{-\tau^2} = p^7$$

Note that $2^{-\alpha} = p^{-2^{-\alpha}}$ and $2^{\xi_1} = r^{2^{-\alpha}}$ for $i \geq 3$. Now we use inequalities (24) and (25) to prove that parameters $Z^{(i)}$ rapidly decline:

$$Z^{(i)} \leq p^{r_i}, \quad t_i = r^{2^{-\alpha}}$$

Indeed, $Z^{(2)}$ satisfies (27). We take $Z^{(i-1)} \leq p^{r_i-1}$ and use induction on the $i$-th segment $1^{r_i}0^{\xi_i}$. Then inequalities (24) and (25) give

$$\ln G^{(i)} \geq -2^{-\xi_i}[p^{r_i+1} + p^{2\xi_i-1}] \geq -2^{r_i}p^{r_i-1}$$

$$\log Z^{(i)} \leq \frac{1}{2} \log[-\ln G^{(i)}] \leq \frac{1}{2} (t_{i-1} - r_i) \log p$$

Note that $r_i = o(t_{i-1})$. Thus, $\log Z^{(i)} \leq s_i \log p$, where

$$s_i = t_{i-1}/r = r^{2^{-\alpha}}(1 - o(t_{i-1}))$$

Then

$$\log Z^{(i)} = 2^{r_i} \log Z^{(i)} \leq 2^{-r_i} s_i \log p = t_i \log p$$

This proves (27) and gives $P_o \leq Z^{(s)}$ for each path $\eta$.

Discussion. Inequalities (28) and (29) show that the initial chains $1^{r_1}$ and the subsequent chains $0^{\xi_i}$ affect parameters $Z^{(i)}$ and $Z^{(i)}$ in a very different way. In particular, (28) shows that any chain $1^{r_1}$ reduces the previous exponential order $t_{i-1} = \log p Z^{(i-1)}$ to $t_{i-1}/2 - o(t_{i-1})$. By contrast, the offset $0^{\xi_i}$ increases this order above $2^{r_i}(t_{i-1}/r)$. For this reason, good BSC(p) with $p \to 0$ may overcompensate long chains $1^{r_i}$ of degrading channels with the much shorter chains $0^{\xi_i}$ of upgrading channels. Note also that equalities (13) and (14) are critical in our proof since they give exact estimates $G^{(i)}$ and $Z^{(i)}$ in all intermediate steps of the segments $1^{r_i}$ or $0^{\xi_i}$, without any loss in performance. To this end, note that inequalities (11) and (12) alone cannot furnish Lemma 6. For example, inequalities (11) replace estimate (26) with a loose bound $Z^{(i)} \sim 1 - e^{-1/32r_{\xi_i}}$. This bound will require a much longer path $0^{\xi_i}$ to achieve a low quantity $Z^{(2)}$, which in turn increases redundancies $r_1$ and $r_2$ above the bound (12) of the weakly optimal codes.

However, this particular construction fails to give the optimal redundancy $r_{opt} \sim p \log 1/p$ or even reduce $r_2$ to the order of $cp \log 1/p$ for some constant $c > 0$. Nor is it known if other low-complexity algorithms for polar or other codes can achieve $r_{opt}$ for $p \to 0$. Note also that the single-boundary set $\eta(L)$ of Lemma 6 does not form an optimized polar code since many other paths $\eta$ also have a vanishing output error rate. For example, any initial segment $1^{r_1}$ of length $r < r_1$ gives rise to many paths $\eta \notin \eta(L)$. To reduce redundancy $r_2$, one may consider a growing set $\{\xi \}$ of boundary paths $\xi$ and form an entire "envelope" of the descendant paths $\eta(\xi)$. Calculating the redundancy for this envelope-type boundary is another open problem that may be related to the Young diagrams.

VI. LOW-RATE CODES WITH A STEPPED BOUNDARY

Consider a sequence of the BSCs(p) with $p = (1 - \epsilon)/2$, where $\epsilon \to 0$ as length $n \to \infty$. Below we study capacity-achieving (CA) codes of rate $R \sim C$ for the case of a vanishing capacity $C = 1 - h(p) \sim \epsilon^2/\ln 4$. It is proved
Lemma 6. Given the same length estimate parameters $Z$.

Note that

Proof. We take $Z(i) \lesssim 2^{\ell_i-1} \log Z(i-1)$ which gives \((35)\) and proves the theorem.

We will now combine codes $R(1, \mu)$ with the high-rate polar codes of Lemma 7 to obtain new CA codes. Note that code $R(1, \mu)$ is defined by a boundary path $\xi(0) = 1^0(0)^{\mu-1}$. We then combine $\xi(0)$ with the boundary $L_c$ of \((31)\) and obtain the extended boundary

$$L_{ext} = \{r_0 = 1, \ell_0 = \mu - 1, L_c\}$$

Lemma 4 shows that $L_{ext}$ generates the direct product $R_{ext}$ of $s + 1$ RM codes $R(r_i, m_i)$. Thus, code $R_{ext}$ has code rate $R$ and length $N$, where

$$R = R(1, \mu)R_{L_c} \sim (\mu + 1)/2^\mu$$

$$N = km, \quad k = 2^\mu, \quad n = 2^m$$

Codes $R_{ext}$ also represent a simple concatenated construction, which first uses $\mu + 1$ arbitrary codewords of the code $R(L_c)$ and forms an $(\mu + 1) \times n$ matrix. Then each column of this matrix is encoded into the code $R(1, \mu)$. The result is an $k \times n$ matrix, which represents a codeword formed by the inner code of length $k$ and $s$ outer codes of length $n$. Below we take $\mu, m \to \infty$. Below we take $p_0 = 2^{-m \theta}$ in \((31)\).

Theorem 8. Let codes $R_{ext}$ of code rate $C(1 - \theta)$ with an $s$-step boundary \((36)\) be used on a BSC($p_\eta$) of capacity $C \to 0$. For any $\theta \in (0, 1)$, codes $R_{ext}$ have decoding complexity $O(N \log N)$ in length $N = nk$ and achieve a bit error probability $P_\eta$ such that

$$\log P_\eta \lesssim -2^{2-s} p_\eta^{c/(c+1)}$$

Proof. Decoding codes $R_{ext}$ can be expressed as SCD; below we also describe it as concatenated decoding of inner codes. We take codes $R(1, \mu)$ of rate $\mu^2/\ln 4$ as $\theta \to 0$. Then $R_{ext} \sim (1 - \theta) \epsilon^2/\ln 4$ as $m \to \infty$. Given a received $2^\mu \times 2^m$ matrix, we first perform ML decoding of each column of length $2^\mu$ into the code $R(1, \mu)$. The resulting $(\mu + 1) \times 2^m$ matrix contains errors with probability $p_0$, or less. Each row is decoded into the code $R(L)$ using SCD on a BSC($p_0$). Note that for any $\theta \in (0, 1)$,

$$m = (2^\mu - 1)(c + 1) \mu \theta - c_1$$

Then \((33)\) gives the bit error rate \((37)\):

$$\log P_\eta \lesssim -2^{2-s} p_\eta^{c/(c+1)}$$

Thus, codes $R_{ext}$ are CA codes. Inner and outer decodings have the complexity $nk \log k$ and $\mu \log n$ bounded by $N \log N$.

Discussion. According to \((39)\), the order $log P_\eta$ depends exponentially on the margin $\theta$ between the code rate $R$ and channel capacity $C$. Below, we compare the performance of codes $R(1, \mu)$ and $R_{ext}$ for the same code rate $R \sim C(1 - \theta)$, and define the minimum code length $k$ or $N$ that enables a given output bit error $P$. Here we consider an asymptotic case with parameters $c \to 1$ and $P \to 0$. For codes $R(1, \mu)$, we have $k \sim P^{-1/\theta}$. For codes $R_{ext}$, we use notation.
For this reason, these single-boundary codes can be considered degrading channels, with a growing length of each segment. Length stretches of the upgrading-degrading channels. and consider the explicit constructions that admit the finite-length stretches of the upgrading-degrading channels. 

$$A = 2^s \theta.$$ Then parameters \((38)\) and \((39)\) yield asymptotic approximations

$$n = 2^m \times k^2 A, \log P \sim -n^{1/2} \sim -k^A \quad (40)$$

(here \(f \approx r \) if \(\log f \sim \log r \)). Recall that the outer codes \(R(C)\) require a vanishing input error rate \(k^{-\theta}\), in which case \(k = B^{1/\theta}\) for some \(B \to \infty\). Then \(N = k^{2A+1} \times B^{1/\theta} (\log^2 P)\). Thus, codes \(R_{ext}\) can improve the trade-off \(k \sim P^{-1/\theta}\) of the inner codes \(R(1, \mu)\) only for the declining error rates \(P = o(1)\). We further note that this is the case for all other known concatenated constructions. In particular, consider a classic concatenation that uses the inner codes \(R(1, \mu)\) and the outer RS codes of the same length \(2^m\) and code rate \(R\) \(- 1\). It can be verified that this construction requires the overall length \(N_1 \approx \max \{\theta^2 \log^2 P, B^{2/\theta}\}\) given the same inner length \(k = B^{1/\theta}\). One possible advantage of codes \(R_{ext}\) over classic concatenation is the extra parameter \(s\) that allows the outer code length \(n\) arbitrarily exceed the inner length \(k\) in \((38)\). In particular, we have inequality \(N \lesssim N_1\) for both cases \(B^{1/\theta} < \log^2 P\) and \(B^{1/\theta} > \log^2 P\). Thus, construction of Theorem 8 allows us to shorten the length \(N_1\) of the classical concatenated construction. More generally, it is an important problem to find low-complexity codes of code rate \(R \to 0\) that can achieve the vanishing error rates at the shorter lengths of order \(N \sim 2^{c/\theta}\) for some \(c \in (0, 1)\).

VII. CONCLUDING REMARKS

In this paper, we address explicit constructions of polar codes that are nearly optimal for the extreme cases of a BSC(\(p\)) with \(p \to 0\) and \(p \to 1/2\). In case of \(p \to 0\), we obtain weakly optimal codes of rate \(R \to 1\), whose redundancy order \(\log P\) declines at the optimal rate \((2)\). For the low-rate codes, we obtain the optimal decline of code rate \(R \to 0\). These simple constructions are completely defined by a single \(s\)-step boundary path \(\xi(C)\) that only depends on transition error probability \(p\). In turn, this definition defines all other paths \(\eta\), which form other sequences of upgrading-degrading channels included in code construction. An important point is that the boundary \(L\) consists of the consecutive chains of upgrading or degrading channels, with a growing length of each segment. For this reason, these single-boundary codes can be considered as direct products of \(s\) Reed-Muller codes. One way to amplify this design is to consider polar codes that include multiple overlapping boundaries \(L_1, \ldots, L_k\) and admit all descendant paths \(\eta\) that satisfy at least one boundary restriction. Another interesting problem is to extend this design to other code rates and consider the explicit constructions that admit the finite-length stretches of the upgrading-degrading channels.

REFERENCES

[1] F.J. MacWilliams and N.J.A. Sloane, “The Theory of Error-Correcting Codes,” North-Holland, Amsterdam, 1981.
[2] I. Dumer, “Recursive decoding and its performance for low-rate Reed-Muller codes,” IEEE Trans. Info. Theory, vol. 50, pp. 811-823, May 2004.
[3] E. Arikan, “Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels,” IEEE Trans. Info. Theory, vol. 55, pp. 3051-3073, July 2009.
[4] R. Blahut, “Algebraic codes for data transmission,” Cambridge Univ. Press, Cambridge, UK, 2003.
[5] I. Dumer and K. Shabunov, “Near-optimum decoding for subcodes of Reed-Muller codes,” 2001 IEEE Intern. Symp. Info. Theory, Washington DC, USA, June 24-29, 2001, p. 329.
[6] I. Dumer and K. Shabunov, “Soft decision decoding of Reed-Muller codes: recursive lists,” IEEE Trans. Info. Theory, vol. 52, pp. 1260-1266, March 2006.
[7] S. B. Korada, “Polar Codes for Channel and Source Coding,” Ph.D. thesis, ´Ecole Polytechnique Federale De Lausanne, 2009.
[8] C. Sch¨{u}r, “A Partial Order For The Synthesized Channels of a Polar Code,” 2016 IEEE Intern. Symp. Info. Theory (ISIT 2016), Barcelona, Spain, July 1-5, 2016, pp. 220-224.
[9] M. Bardet, V. Dragoi, A. Ottmani, and J.-P. Tillich, “Algebraic properties of polar codes from a new polynomial formalism,” 2016 IEEE Intern. Symp. Info. Theory (ISIT 2016), Barcelona, Spain, July 1-5, 2016, pp. 230-234.
[10] M. Burnashev and I. Dumer, “Error Exponents for Recursive Decoding of Reed-Muller Codes on a Binary-Symmetric Channel,” IEEE Trans. Info. Theory, 52, 11, pp. 4880-4891, 2006.
[11] V. Gurusswami and P. Xia, “Polar Codes: Speed of Polarization and Polynomial Gap to Capacity,” IEEE Trans. Info. Theory, vol. 61, pp. 3-16, Jan. 2015.
[12] R. Saptharshi, A. Shpilka and B.L. Volk, “Efficiently decoding Reed-Muller codes from random errors,” Proc. 48th STOC ‘16, pp. 227-235, Cambridge, MA, USA, June 19, 2016.
[13] E. Abbe, A. Shpilka, and A. Wigderson, “Reed-Muller Codes for Random Erasures and Errors,” Proc. 47th Symp. Theory of Comp. (STOC ‘15), pp. 297–306, Portland, OR, USA, June 15, 2015.
[14] V. Sid´el’nikov and A. Pershakov, “Decoding of Reed-Muller codes with a large number of errors,” Probl. Info. Transmission, vol. 28, no. 3, pp. 80-94, 1992.
[15] T.J. Richardson, M.A. Shokrollahi, and R.L. Urbanke, “Design of capacity-approaching irregular low-density parity-check codes,” IEEE Trans. Inform. Theory, vol. 47, no. 2, pp. 619-637, 2001.

Appendix. Proof of Lemma 3. To prove Lemma 3 we will assume that any channel \(W_X\) satisfies the “symmetry” condition \((15)\), p. 628. This condition (expressed in terms of log likelihoods in \((15)\) implies that the likelihoods \(h\) of transmitted symbols have the probability density function (pdf) \(p(x) \equiv p_x(x)\) such that

$$p(x)/p(x^{-1}) = x, \quad \forall x \in (0, \infty). \quad (41)$$

Condition \((41)\) can be used for many conventional channels; in particular, for a BSC(\(p\)) or an AWGN channel. It is also proven in \((15)\) that the “symmetry” condition is left intact by transformations \((9)\). Namely, both rv \(h_{ij}^{(0)}\) and \(h_{ij}^{(1)}\) in \((9)\) satisfy condition \((41)\) if so do rv \(h_{0j}\) and \(h_{1j}\).

Next, we consider an output \(h(a)\) of the prefix path \(a\). Let \(h_1, h_2, h_3,\) and \(h_4\) denote 4 independent ID rv, which represent 4 different outputs \(h(a)\) of the prefix \(a\). We need to calculate the outputs \(h_{00} = h(a01)\) and \(h_{10} = h(a10)\) and prove that

$$E h_{00}^{\lambda} \leq E h_{10}^{\lambda} \quad \text{if } \lambda \in [0, 1].$$

An equivalent formulation is to prove inequality \(E f_{a01}^{\lambda} \leq E f_{10}^{\lambda}\) given inverse likelihoods \(f_i = h^{-1}\), \(f_{01} = h_{01}\) and \(f_{10} = h_{10}\). Correspondingly, we consider the 4-dimensional space \(R_+^4\) formed by vectors \(F = (f_1, f_2, f_3, f_4)\) with positive coordinates. For extended subpaths \(a01\) and \(a10\), recalculation \((9)\) give the rv outputs

$$f_{00} = f_1 f_2 f_3 f_4, \quad f_{10} = (f_1 + f_2)(f_3 + f_4).$$

Below we also consider another rv

$$u_{01} = (f_1 + f_2)(f_3 + f_4)/2(1 + f_1 f_2 f_3 f_4)$$
and prove two inequalities
\[ E f_{01}^\lambda \leq E u_{01}^\lambda \leq E f_{10}^\lambda, \quad \lambda \in [0, 1]. \]  
(42)

To prove the left inequality, note that \( u_{01} = (f'_{01} + f''_{01})/2 \), where
\[ f'_{01} = \frac{f_1 f_3 + f_2 f_4}{1 + f_1 f_2 f_3 f_4}, \quad f''_{01} = \frac{f_1 f_4 + f_2 f_3}{1 + f_1 f_2 f_3 f_4}. \]
The variables \( f'_{01} \) and \( f''_{01} \) are obtained from \( f_{01} \) by replacements \( f_2 \leftrightarrow f_3 \) and \( f_2 \leftrightarrow f_4 \) respectively. Then independent and ID rv \( f_1 \) give equalities
\[ E f_{01}^\lambda = E (f'_{01})^\lambda = E (f''_{01})^\lambda. \]

Since \( x^\lambda \) is a concave function of any \( x > 0 \) for \( \lambda \in [0, 1] \),
\[ \frac{(f'_{01})^\lambda}{2} + \frac{(f''_{01})^\lambda}{2} \leq \left( \frac{f'_{01} + f''_{01}}{2} \right)^\lambda = u_{01}^\lambda \]  
(43)

and \( E f_{01}^\lambda \leq E u_{01}^\lambda \).

To compare the expectations \( E u_{01}^\lambda \) and \( E f_{10}^\lambda \), we combine each vector \( F \equiv F_0 \in \mathbb{R}_+^4 \) with three other vectors (which may also coincide with \( F \)):
\[ F_1 = (f_1^{-1}, f_2^{-1}, f_3, f_4), \quad F_2 = (f_1^{-1}, f_2^{-1}, f_3^{-1}, f_4), \]
\[ F_3 = (f_1, f_2, f_3^{-1}, f_4^{-1}). \]

We also consider the orbit \( T = \{F_0, F_1, F_2, F_3\} \) of vector \( F_0 \in \mathbb{R}_+^4 \). Clearly, the whole space \( \mathbb{R}_+^4 \) is now partitioned into non-intersecting orbits \( T \). Below we use notation
\[ \alpha = f_1 f_2, \quad \beta = f_3 f_4, \quad A = (f_1 + f_2)(f_3 + f_4). \]

It can be readily verified that the rv \( f_{10}(T) \) does not change on the orbit \( T \):
\[ f_{10}(F_i) = \frac{A}{(1 + \alpha)(1 + \beta)}, \quad i = 0, ..., 3, \]  
(44)

while \( u_{01}(T) \) takes two values
\[ u_{01}(F_0) = u_{01}(F_2) = \frac{A}{2(1 + \alpha \beta)}, \]
\[ u_{01}(F_1) = u_{01}(F_3) = \frac{A}{2(\alpha + \beta)}. \]

Let \( p = p(F_0) \) denote the pdf of the 4-dimensional rv \( F_0 \in \mathbb{R}_+^4 \), which consists of inverse likelihoods. According to (41), the pdfs of other orbit points are
\[ p(F_2) = \alpha \beta p, \quad p(F_1) = \alpha p, \quad p(F_3) = \beta p \]  
(45)

Then simple recalculations using equalities (44) and (45) give
\[ E f_{10}^\lambda(T) = p A^\lambda \left[(1 + \alpha)(1 + \beta)\right]^{1-\lambda}, \]
\[ E u_{01}^\lambda(T) = p A^\lambda 2^{-\lambda} \left[(1 + \alpha \beta)^{1-\lambda} + (\alpha + \beta)^{1-\lambda}\right]. \]

Since \( x^{1-\lambda} \) is a concave function, we have inequality
\[ \frac{(1+\alpha\beta)^{1-\lambda}}{2} + \frac{(\alpha+\beta)^{1-\lambda}}{2} \leq \left[\frac{(1+\alpha)(1+\beta)}{2}\right]^{1-\lambda} \]  
(46)

which proves the right inequality in (42). The second case with \( \lambda \in [1, \infty) \) is studied similarly. Now both \( x^\lambda \) and \( x^{1-\lambda} \) are convex functions of \( x > 0 \). Then inequalities (43) and (46) change their sign and we have inequality (17). \( \square \)