Erdal Arıkan

Polar Coding

ISIT 2012 Tutorial

June 27, 2012

Preface

These notes on polar coding are prepared for a tutorial to be given at ISIT 2012. The notes are based on the author's paper "Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels," published in the July 2009 issue of the IEEE Transactions on Information Theory. The 2009 paper has been updated to cover two major advances that took place since the publication of that paper: exponential error bounds for polar codes and an efficient algorithm for constructing polar codes. Both of these topics are now an integral part of the core theory of polar coding. In its present form, these notes present the basic theory of polarization and polar coding in a fairly complete manner. There have been many more important advances in polar coding in the few years since the subject appeared: non-binary polarization, source polarization, multi-terminal polarization, polarization under memory, quantum polar coding, to name some. Also a large number of papers exist now on practical aspects of polar coding and their potential for applications. These subjects are not covered in these notes since the goal has been to present the basic theory within the confines of a three-hour tutorial.

Ankara, *E. Arıkan*
June 2012

Contents

Chapter 0
Preliminaries and Notation

Abstract This chapter gathers the notation and some basic facts that are used throughout.

0.1 Notation

We denote random variables (RVs) by upper-case letters, such as X, Y, and their realizations (sample values) by the corresponding lower-case letters, such as x, y. For X a RV, P_X denotes the probability assignment on X. For a joint ensemble of RVs (X,Y), $P_{X,Y}$ denotes the joint probability assignment. We use the standard notation $I(X;Y)$, $I(X;Y|Z)$ to denote the mutual information and its conditional form, respectively.

We use the notation a_1^N as shorthand for denoting a row vector $(a_1,\ldots,a_N)$. Given such a vector a_1^N, we write a_i^j, $1 \le i, j \le N$, to denote the subvector $(a_i,\ldots,a_j)$; if $j < i$, a_i^j is regarded as void. Given a_1^N and $\mathscr{A} \subset \{1,\ldots,N\}$, we write $a_{\mathscr{A}}$ to denote the subvector $(a_i : i \in \mathscr{A})$. We write $a_{1,o}^j$ to denote the subvector with odd indices $(a_k : 1 \le k \le j;\ k \text{ odd})$. We write $a_{1,e}^j$ to denote the subvector with even indices $(a_k : 1 \le k \le j;\ k \text{ even})$. For example, for $a_1^5 = (5,4,6,2,1)$, we have $a_2^4 = (4,6,2)$, $a_{1,e}^5 = (4,2)$, $a_{1,o}^4 = (5,6)$. The notation 0_1^N is used to denote the all-zero vector.

Code constructions in these notes will be carried out in vector spaces over the binary field GF(2). Unless specified otherwise, all vectors, matrices, and operations on them will be over GF(2). In particular, for a_1^N, b_1^N vectors over GF(2), we write $a_1^N \oplus b_1^N$ to denote their componentwise mod-2 sum. The Kronecker product of an m-by-n matrix $A = [A_{ij}]$ and an r-by-s matrix $B = [B_{ij}]$ is defined as

$$A \otimes B = \begin{bmatrix} A_{11}B & \cdots & A_{1n}B \\ \vdots & \ddots & \vdots \\ A_{m1}B & \cdots & A_{mn}B \end{bmatrix},$$

which is an *mr*-by-*ns* matrix. The Kronecker power $A^{\otimes n}$ is defined as $A \otimes A^{\otimes(n-1)}$ for all $n \geq 1$. We will follow the convention that $A^{\otimes 0} \overset{\Delta}{=} [1]$.

We write $|\mathscr{A}|$ to denote the number of elements in a set $\mathscr{A}$. We write $1_{\mathscr{A}}$ to denote the indicator function of a set $\mathscr{A}$; thus, $1_{\mathscr{A}}(x)$ equals 1 if $x \in \mathscr{A}$ and 0 otherwise.

We use the standard Landau notation $O(N)$, $o(N)$, $\omega(N)$ to denote the asymptotic behavior of functions.

Throughout log will denote logarithm to the base 2. The unit for channel capacities and code rates will be *bits*.

0.2 Binary Channels and Symmetric Capacity

We write $W : \mathscr{X} \to \mathscr{Y}$ to denote a generic binary-input discrete memoryless channel (B-DMC) with input alphabet $\mathscr{X}$, output alphabet $\mathscr{Y}$, and transition probabilities $W(y|x)$, $x \in \mathscr{X}$, $y \in \mathscr{Y}$. The input alphabet $\mathscr{X}$ will always be $\{0,1\}$, the output alphabet and the transition probabilities may be arbitrary. We write W^N to denote the channel corresponding to N uses of W; thus, $W^N : \mathscr{X}^N \to \mathscr{Y}^N$ with $W^N(y_1^N \mid x_1^N) = \prod_{i=1}^N W(y_i \mid x_i)$.

The symmetric capacity of a B-DMC W is defined as

$$I(W) \overset{\Delta}{=} \sum_{y\in\mathscr{Y}} \sum_{x\in\mathscr{X}} \frac{1}{2} W(y|x) \log \frac{W(y|x)}{\frac{1}{2}W(y|0) + \frac{1}{2}W(y|1)}$$

Since we use base-2 logarithms, $I(W)$ takes values in $[0,1]$ and is measured in bits.

The symmetric capacity $I(W)$ is the highest rate at which reliable communication is possible across W using the inputs of W with equal frequency. It equals the Shannon capacity when W is a *symmetric* channel, i.e., a channel for which there exists a permutation π of the output alphabet $\mathscr{Y}$ such that (i) $\pi^{-1} = \pi$ and (ii) $W(y|1) = W(\pi(y)|0)$ for all $y \in \mathscr{Y}$.

The binary symmetric channel (BSC) and the binary erasure channel (BEC) are examples of symmetric channels. A BSC is a B-DMC W with $\mathscr{Y} = \{0,1\}$, $W(0|0) = W(1|1)$, and $W(1|0) = W(0|1)$. A B-DMC W is called a BEC if for each $y \in \mathscr{Y}$, either $W(y|0)W(y|1) = 0$ or $W(y|0) = W(y|1)$. In the latter case, y is said to be an *erasure* symbol. The sum of $W(y|0)$ over all erasure symbols y is called the erasure probability of the BEC.

0.3 Channel Bhattacharyya parameter: A measure of reliability

The Bhattacharyya parameter of a B-DMC W is defined as

$$Z(W) \overset{\Delta}{=} \sum_{y\in\mathscr{Y}} \sqrt{W(y|0)W(y|1)}.$$

The Bhattacharyya parameter $Z(W)$ is an upper bound on the probability of MAP decision error when W is used only once to transmit a single bit, a-priori equally likely to be 0 or 1. Hence, $Z(W)$ serves as a measures of *reliability* for W. It is easy to see that $Z(W)$ takes values in $[0,1]$.

Intuitively, one would expect that $I(W) \approx 1$ iff $Z(W) \approx 0$, and $I(W) \approx 0$ iff $Z(W) \approx 1$. The following bounds make this precise.

Proposition 1 *For any B-DMC W, we have*

$$I(W) \geq \log \frac{2}{1+Z(W)}, \tag{0.1}$$

$$I(W) \leq \sqrt{1-Z(W)^2}. \tag{0.2}$$

Furthermore,

$$I(W)+Z(W) \geq 1 \tag{0.3}$$

with equality iff W is a BEC.

Proof of inequality (0.1)*:*
This is proved easily by noting that

$$\log \frac{2}{1+Z(W)}$$

actually equals the channel parameter denoted by $E_0(1,Q)$ by Gallager [6, Section 5.6] with Q taken as the uniform input distribution. (This parameter may be called the *symmetric cutoff rate* of the channel.) It is well known (and shown in the same section of [6]) that $I(W) \geq E_0(1,Q)$. This proves (0.1).

Proof of inequality (0.2)*:*
For any B-DMC $W : \mathscr{X} \to \mathscr{Y}$, define

$$d(W) \overset{\Delta}{=} \frac{1}{2} \sum_{y \in \mathscr{Y}} |W(y|0) - W(y|1)|.$$

This is the variational distance between the two distributions $W(y|0)$ and $W(y|1)$ over $y \in \mathscr{Y}$.

Lemma 1 *For any B-DMC W, $I(W) \leq d(W)$.*

Proof. Let W be an arbitrary B-DMC with output alphabet $\mathscr{Y} = \{1,\dots,n\}$ and put $P_i = W(i|0)$, $Q_i = W(i|1)$, $i = 1,\dots,n$. By definition,

$$I(W) = \sum_{i=1}^{n} \frac{1}{2} \left[P_i \log \frac{P_i}{\frac{1}{2}P_i + \frac{1}{2}Q_i} + Q_i \log \frac{Q_i}{\frac{1}{2}P_i + \frac{1}{2}Q_i} \right].$$

The *i*th bracketed term under the summation is given by

$$f(x) \overset{\Delta}{=} x \log \frac{x}{x+\delta} + (x+2\delta)\log \frac{x+2\delta}{x+\delta}$$

where $x = \min\{P_i, Q_i\}$ and $\delta = \frac{1}{2}|P_i - Q_i|$. We now consider maximizing $f(x)$ over $0 \le x \le 1-2\delta$. We compute

$$\frac{df}{dx} = \frac{1}{2}\log \frac{\sqrt{x(x+2\delta)}}{(x+\delta)}$$

and recognize that $\sqrt{x(x+2\delta)}$ and $(x+\delta)$ are, respectively, the geometric and arithmetic means of the numbers x and $(x+2\delta)$. So, $df/dx \le 0$ and $f(x)$ is maximized at $x=0$, giving the inequality $f(x) \le 2\delta$. Using this in the expression for $I(W)$, we obtain the claim of the lemma,

$$I(W) \le \sum_{i=1} \frac{1}{2}|P_i - Q_i| = d(W).$$

Lemma 2 *For any B-DMC* W, $d(W) \le \sqrt{1-Z(W)^2}$.

Proof. Let W be an arbitrary B-DMC with output alphabet $\mathscr{Y} = \{1,\dots,n\}$ and put $P_i = W(i|0)$, $Q_i = W(i|1)$, $i = 1,\dots,n$. Let $\delta_i \overset{\Delta}{=} \frac{1}{2}|P_i - Q_i|$, $\delta \overset{\Delta}{=} d(W) = \sum_{i=1}^n \delta_i$, and $R_i \overset{\Delta}{=} (P_i + Q_i)/2$. Then, we have $Z(W) = \sum_{i=1}^n \sqrt{(R_i - \delta_i)(R_i + \delta_i)}$. Clearly, $Z(W)$ is upper-bounded by the maximum of $\sum_{i=1}^n \sqrt{R_i^2 - \delta_i^2}$ over $\{\delta_i\}$ subject to the constraints that $0 \le \delta_i \le R_i$, $i = 1,\dots,n$, and $\sum_{i=1}^n \delta_i = \delta$. To carry out this maximization, we compute the partial derivatives of $Z(W)$ with respect to δ_i,

$$\frac{\partial Z}{\partial \delta_i} = -\frac{\delta_i}{\sqrt{R_i^2 - \delta_i^2}}, \qquad \frac{\partial^2 Z}{\partial \delta_i^2} = -\frac{R_i^2}{\sqrt[3/2]{R_i^2 - \delta_i^2}},$$

and observe that $Z(W)$ is a decreasing, concave function of δ_i for each i, within the range $0 \le \delta_i \le R_i$. The maximum occurs at the solution of the set of equations $\partial Z/\partial \delta_i = k$, all i, where k is a constant, i.e., at $\delta_i = R_i\sqrt{k^2/(1+k^2)}$. Using the constraint $\sum_i \delta_i = \delta$ and the fact that $\sum_{i=1}^n R_i = 1$, we find $\sqrt{k^2/(1+k^2)} = \delta$. So, the maximum occurs at $\delta_i = \delta R_i$ and has the value $\sum_{i=1}^n \sqrt{R_i^2 - \delta^2 R_i^2} = \sqrt{1-\delta^2}$. We have thus shown that $Z(W) \le \sqrt{1-d(W)^2}$, which is equivalent to $d(W) \le \sqrt{1-Z(W)^2}$.

From the above two lemmas, the proof of (0.2) is immediate.

Proof of inequality (0.3)*:* We defer this proof until Chapter 3 where it will follow as a simple corollary to the results there.

It can be seen that inequality 0.3 is stronger than inequality 0.1 and will prove useful later on. The weaker inequality (0.1) is sufficient to develop the polarization results for the time being.

Chapter 1
Overview of Results

Abstract Shannon proved the achievability part of his noisy channel coding theorem using a random-coding argument which showed the existence of capacity-achieving code sequences without exhibiting any specific sequence [15]. Polar codes are an explicit construction that provably achieves channel capacity with low-complexity encoding, decoding, and code construction algorithms. This chapter gives an overview of channel polarization and polar coding.

1.1 Channel polarization

Channel polarization is a transformation by which one manufactures out of N independent copies of a given B-DMC W a second set of N channels $\{W_N^{(i)} : 1 \leq i \leq N\}$ such that, as N becomes large, the symmetric capacity terms $\{I(W_N^{(i)})\}$ tend towards 0 or 1 for all but a vanishing fraction of indices i. The channel polarization operation consists of a channel combining phase and a channel splitting phase.

1.1.1 Channel combining

This phase combines copies of a given B-DMC W in a recursive manner to produce a vector channel $W_N : \mathscr{X}^N \to \mathscr{Y}^N$, where N can be any power of two, $N = 2^n$, $n \geq 0$. The recursion begins at the 0-th level ($n = 0$) with only one copy of W and we set $W_1 \overset{\Delta}{=} W$. The first level ($n = 1$) of the recursion combines two independent copies of W_1 as shown in Fig. 1 and obtains the channel $W_2 : \mathscr{X}^2 \to \mathscr{Y}^2$ with the transition probabilities

$$W_2(y_1,y_2|u_1,u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2). \tag{1.1}$$

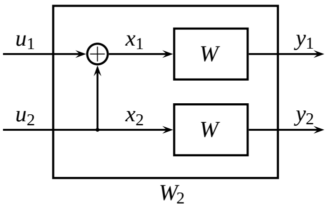

Fig. 1.1 The channel W_2.

The next level of the recursion is shown in Fig. 2 where two independent copies of W_2 are combined to create the channel $W_4 : \mathscr{X}^4 \to \mathscr{Y}^4$ with transition probabilities $W_4(y_1^4|u_1^4) = W_2(y_1^2|u_1 \oplus u_2, u_3 \oplus u_4)W_2(y_3^4|u_2, u_4)$.

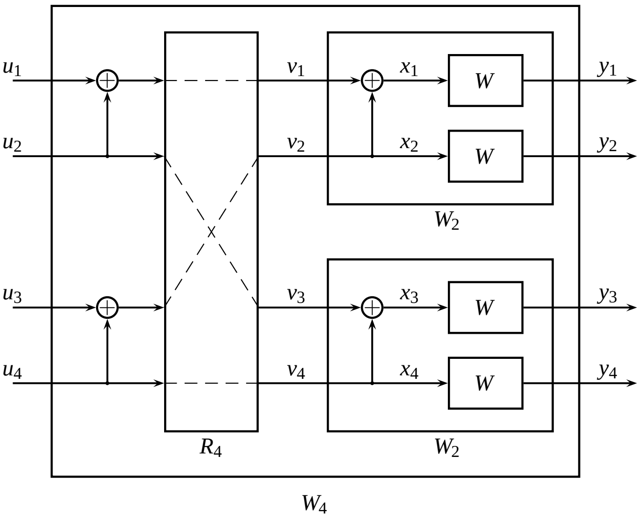

Fig. 1.2 The channel W_4 and its relation to W_2 and W.

In Fig. 2, R_4 is the permutation operation that maps an input (s_1, s_2, s_3, s_4) to $v_1^4 = (s_1, s_3, s_2, s_4)$. The mapping $u_1^4 \mapsto x_1^4$ from the input of W_4 to the input of W^4 can be written as $x_1^4 = u_1^4 G_4$ with $G_4 = \begin{bmatrix} 1&0&0&0 \\ 1&0&1&0 \\ 1&1&0&0 \\ 1&1&1&1 \end{bmatrix}$. Thus, we have the relation $W_4(y_1^4|u_1^4) = W^4(y_1^4|u_1^4 G_4)$ between the transition probabilities of W_4 and those of W^4.

The general form of the recursion is shown in Fig. 3 where two independent copies of $W_{N/2}$ are combined to produce the channel W_N. The input vector u_1^N to W_N is first transformed into s_1^N so that $s_{2i-1} = u_{2i-1} \oplus u_{2i}$ and $s_{2i} = u_{2i}$ for $1 \leq i \leq$

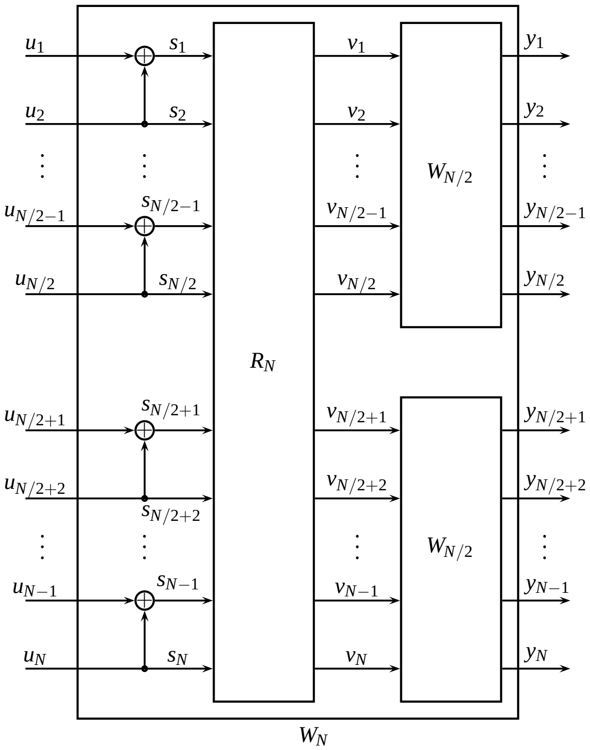

Fig. 1.3 Recursive construction of W_N from two copies of $W_{N/2}$.

$N/2$. The operator R_N in the figure is a permutation, known as the *reverse shuffle* operation, and acts on its input $s_1^N$to produce $v_1^N = (s_1, s_3, \ldots, s_{N-1}, s_2, s_4, \ldots, s_N)$, which becomes the input to the two copies of $W_{N/2}$ as shown in the figure.

We observe that the mapping $u_1^N \mapsto v_1^N$ is linear over GF(2). It follows by induction that the overall mapping $u_1^N \mapsto x_1^N$, from the input of the synthesized channel W_N to the input of the underlying raw channels W^N, is also linear and may be represented by a matrix G_N so that $x_1^N = u_1^N G_N$. We call G_N the *generator matrix* of size N. The transition probabilities of the two channels W_N and W^N are related by

$$W_N(y_1^N|u_1^N) = W^N(y_1^N|u_1^N G_N) \tag{1.2}$$

for all $y_1^N \in \mathscr{Y}^N$, $u_1^N \in \mathscr{X}^N$. We will show in Sect. 5.1 that G_N equals $B_N F^{\otimes n}$ for any $N = 2^n$, $n \geq 0$, where B_N is a permutation matrix known as *bit-reversal* and $F \overset{\Delta}{=} \left[\begin{smallmatrix} 1 & 0 \\ 1 & 1 \end{smallmatrix}\right]$. Note that the channel combining operation is fully specified by the matrix F. Also note that G_N and $F^{\otimes n}$ have the same set of rows, but in a different (bit-reversed) order; we will discuss this topic more fully in Sect. 5.1.

1.1.2 Channel splitting

Having synthesized the vector channel W_N out of W^N, the next step of channel polarization is to split W_N back into a set of N binary-input coordinate channels $W_N^{(i)} : \mathscr{X} \to \mathscr{Y}^N \times \mathscr{X}^{i-1}$, $1 \leq i \leq N$, defined by the transition probabilities

$$W_N^{(i)}(y_1^N, u_1^{i-1}|u_i) \overset{\Delta}{=} \sum_{u_{i+1}^N \in \mathscr{X}^{N-i}} \frac{1}{2^{N-1}} W_N(y_1^N|u_1^N), \tag{1.3}$$

where (y_1^N, u_1^{i-1}) denotes the output of $W_N^{(i)}$ and u_i its input.

To gain an intuitive understanding of the channels $\{W_N^{(i)}\}$, consider a genie-aided successive cancellation decoder in which the ith decision element estimates u_i after observing y_1^N and the *past* channel inputs u_1^{i-1} (supplied correctly by the genie regardless of any decision errors at earlier stages). If u_1^N is a-priori uniform on $\mathscr{X}^N$, then $W_N^{(i)}$ is the effective channel seen by the ith decision element in this scenario.

1.1.3 Channel polarization

Theorem 1 *For any B-DMC W, the channels $\{W_N^{(i)}\}$ polarize in the sense that, for any fixed $\delta \in (0,1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1,\dots,N\}$ for which $I(W_N^{(i)}) \in (1-\delta, 1]$ goes to $I(W)$ and the fraction for which $I(W_N^{(i)}) \in [0,\delta)$ goes to $1 - I(W)$.*

This theorem is proved in Sect. 3.3.

The polarization effect is illustrated in Fig. 4 for W a BEC with erasure probability $\varepsilon = 0.5$. The numbers $\{I(W_N^{(i)})\}$ have been computed using the recursive relations

$$\begin{aligned} I(W_N^{(2i-1)}) &= I(W_{N/2}^{(i)})^2, \\ I(W_N^{(2i)}) &= 2I(W_{N/2}^{(i)}) - I(W_{N/2}^{(i)})^2, \end{aligned} \tag{1.4}$$

with $I(W_1^{(1)}) = 1 - \varepsilon$. This recursion is valid only for BECs and it is proved in Sect. 2.2. Figure 4 shows that $I(W^{(i)})$ tends to be near 0 for small i and near 1 for large i. However, $I(W_N^{(i)})$ shows an erratic behavior for an intermediate range of i.

For general B-DMCs, the calculation of $I(W_N^{(i)})$ with sufficient degree of precision is an important problem for constructing polar codes. This issue is discussed in Sect. 5.3.

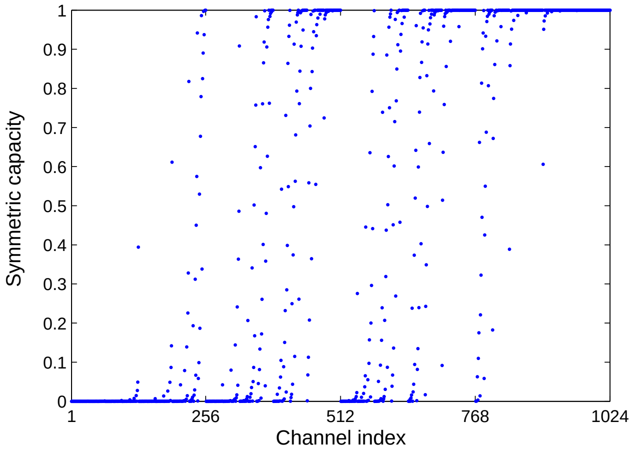

Fig. 1.4 Plot of $I(W_N^{(i)})$ vs. $i = 1,\dots,N = 2^{10}$ for a BEC with $\varepsilon = 0.5$.

1.1.4 Rate of polarization

For proving coding theorems, the speed with which the polarization effect takes hold as a function of N is important. Our main result in this regard is given in terms of the parameters

$$Z(W_N^{(i)}) = \sum_{y_1^N \in \mathscr{Y}^N} \sum_{u_1^{i-1} \in \mathscr{X}^{i-1}} \sqrt{W_N^{(i)}(y_1^N, u_1^{i-1} \mid 0)\, W_N^{(i)}(y_1^N, u_1^{i-1} \mid 1)}. \tag{1.5}$$

Theorem 2 *Let W be a B-DMC. For any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of sets $\{\mathscr{A}_N\}$ such that $\mathscr{A}_N \subset \{1,\dots,N\}$, $|\mathscr{A}_N| \geq NR$, and*

$$\sum_{i \in \mathscr{A}_N} Z(W_N^{(i)}) = o(2^{-N^\beta}). \tag{1.6}$$

Conversely, if $R > 0$ and $\beta > \frac{1}{2}$, then for any sequence of sets $\{\mathscr{A}_N\}$ with $\mathscr{A}_N \subset \{1,\dots,N\}$, $|\mathscr{A}_N| \geq NR$, we have

$$\max\{Z(W_N^{(i)}) : i \in \mathscr{A}_N\} = \omega(2^{-N^\beta}). \tag{1.7}$$

This theorem is proved in Chapter 3.

We stated the polarization result in Theorem 2 in terms $\{Z(W_N^{(i)})\}$ rather than $\{I(W_N^{(i)})\}$ because this form is better suited to the coding results that we will develop. A rate of polarization result in terms of $\{I(W_N^{(i)})\}$ can be obtained from Theorem 2 with the help of Prop. 1.

1.2 Polar coding

Polar coding is a method that takes advantage of the polarization effect to construct codes that achieve the symmetric channel capacity $I(W)$. The basic idea of polar coding is to create a coding system where one can access each coordinate channel $W_N^{(i)}$ individually and send data only through those for which $Z(W_N^{(i)})$ is near 0.

1.2.1 G_N-coset codes

We first describe a class of block codes that contain polar codes—the codes of main interest—as a special case. The block-lengths N for this class are restricted to powers of two, $N = 2^n$ for some $n \geq 0$. For a given N, each code in the class is encoded in the same manner, namely,

$$x_1^N = u_1^N G_N \tag{1.8}$$

where G_N is the generator matrix of order N, defined above. For $\mathscr{A}$ an arbitrary subset of $\{1,\dots,N\}$, we may write (1.8) as

$$x_1^N = u_{\mathscr{A}} G_N(\mathscr{A}) \oplus u_{\mathscr{A}^c} G_N(\mathscr{A}^c) \tag{1.9}$$

where $G_N(\mathscr{A})$ denotes the submatrix of G_N formed by the rows with indices in $\mathscr{A}$.

If we now fix $\mathscr{A}$ and $u_{\mathscr{A}^c}$, but leave $u_{\mathscr{A}}$ as a free variable, we obtain a mapping from source blocks $u_{\mathscr{A}}$ to codeword blocks x_1^N. This mapping is a *coset code*: it is a coset of the linear block code with generator matrix $G_N(\mathscr{A})$, with the coset determined by the fixed vector $u_{\mathscr{A}^c} G_N(\mathscr{A}^c)$. We will refer to this class of codes collectively as *G_N-coset codes*. Individual G_N-coset codes will be identified by a parameter vector $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$, where K is the code dimension and specifies the size of $\mathscr{A}$.[1] The ratio K/N is called the *code rate*. We will refer to $\mathscr{A}$ as the *information set* and to $u_{\mathscr{A}^c} \in \mathscr{X}^{N-K}$ as *frozen* bits or vector.

For example, the $(4,2,\{2,4\},(1,0))$ code has the encoder mapping

$$\begin{aligned} x_1^4 &= u_1^4 G_4 \\ &= (u_2, u_4) \begin{bmatrix} 1\ 0\ 1\ 0 \\ 1\ 1\ 1\ 1 \end{bmatrix} + (1,0) \begin{bmatrix} 1\ 0\ 0\ 0 \\ 1\ 1\ 0\ 0 \end{bmatrix}. \end{aligned} \tag{1.10}$$

[1] We include the redundant parameter K in the parameter set because often we consider an ensemble of codes with K fixed and $\mathscr{A}$ free.

For a source block $(u_2,u_4) = (1,1)$, the coded block is $x_1^4 = (1,1,0,1)$.

Polar codes will be specified shortly by giving a particular rule for the selection of the information set $\mathscr{A}$.

1.2.2 A successive cancellation decoder

Consider a G_N-coset code with parameter $(N,K,\mathscr{A},u_{\mathscr{A}^c})$. Let u_1^N be encoded into a codeword x_1^N, let x_1^N be sent over the channel W^N, and let a channel output y_1^N be received. The decoder's task is to generate an estimate $\hat{u}_1^N$ of u_1^N, given knowledge of $\mathscr{A}$, $u_{\mathscr{A}^c}$, and y_1^N. Since the decoder can avoid errors in the frozen part by setting $\hat{u}_{\mathscr{A}^c} = u_{\mathscr{A}^c}$, the real decoding task is to generate an estimate $\hat{u}_{\mathscr{A}}$ of $u_{\mathscr{A}}$.

The coding results in this paper will be given with respect to a specific successive cancellation (SC) decoder, unless some other decoder is mentioned. Given any $(N,K,\mathscr{A},u_{\mathscr{A}^c})$ G_N-coset code, we will use a SC decoder that generates its decision $\hat{u}_1^N$ by computing

$$\hat{u}_i \overset{\Delta}{=} \begin{cases} u_i, & \text{if } i \in \mathscr{A}^c \\ h_i(y_1^N, \hat{u}_1^{i-1}), & \text{if } i \in \mathscr{A} \end{cases} \tag{1.11}$$

in the order i from 1 to N, where $h_i : \mathscr{Y}^N \times \mathscr{X}^{i-1} \to \mathscr{X}$, $i \in \mathscr{A}$, are *decision functions* defined as

$$h_i(y_1^N, \hat{u}_1^{i-1}) \overset{\Delta}{=} \begin{cases} 0, & \text{if } \frac{W_N^{(i)}(y_1^N,\hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N,\hat{u}_1^{i-1}|1)} \geq 1 \\ 1, & \text{otherwise} \end{cases} \tag{1.12}$$

for all $y_1^N \in \mathscr{Y}^N$, $\hat{u}_1^{i-1} \in \mathscr{X}^{i-1}$. We will say that a decoder *block error* occurred if $\hat{u}_1^N \neq u_1^N$ or equivalently if $\hat{u}_{\mathscr{A}} \neq u_{\mathscr{A}}$.

The decision functions $\{h_i\}$ defined above resemble ML decision functions but are not exactly so, because they treat the *future* frozen bits $(u_j : j > i,\, j \in \mathscr{A}^c)$ as RVs, rather than as known bits. In exchange for this suboptimality, $\{h_i\}$ can be computed efficiently using recursive formulas, as we will show in Sect. 2.1. Apart from algorithmic efficiency, the recursive structure of the decision functions is important because it renders the performance analysis of the decoder tractable. Fortunately, the loss in performance due to not using true ML decision functions happens to be negligible: $I(W)$ is still achievable.

1.2.3 Code performance

The notation $P_e(N,K,\mathscr{A},u_{\mathscr{A}^c})$ will denote the probability of block error for a $(N,K,\mathscr{A},u_{\mathscr{A}^c})$ code, assuming that each data vector $u_{\mathscr{A}} \in \mathscr{X}^K$ is sent with probability 2^{-K} and decoding is done by the above SC decoder. More precisely,

$$P_e(N,K,\mathscr{A},u_{\mathscr{A}^c}) \overset{\Delta}{=} \sum_{u_{\mathscr{A}} \in \mathscr{X}^K} \frac{1}{2^K} \sum_{y_1^N \in \mathscr{Y}^N : \hat{u}_1^N(y_1^N) \neq u_1^N} W_N(y_1^N | u_1^N).$$

The average of $P_e(N,K,\mathscr{A},u_{\mathscr{A}^c})$ over all choices for $u_{\mathscr{A}^c}$ will be denoted by $P_e(N,K,\mathscr{A})$:

$$P_e(N,K,\mathscr{A}) \overset{\Delta}{=} \sum_{u_{\mathscr{A}^c} \in \mathscr{X}^{N-K}} \frac{1}{2^{N-K}} P_e(N,K,\mathscr{A},u_{\mathscr{A}^c}).$$

A key bound on block error probability under SC decoding is the following.

Proposition 2 *For any B-DMC W and any choice of the parameters $(N,K,\mathscr{A})$,*

$$P_e(N,K,\mathscr{A}) \leq \sum_{i \in \mathscr{A}} Z(W_N^{(i)}). \tag{1.13}$$

Hence, for each $(N,K,\mathscr{A})$, there exists a frozen vector $u_{\mathscr{A}^c}$ such that

$$P_e(N,K,\mathscr{A},u_{\mathscr{A}^c}) \leq \sum_{i \in \mathscr{A}} Z(W_N^{(i)}). \tag{1.14}$$

This is proved in Sect. 4.3. This result suggests choosing $\mathscr{A}$ from among all K-subsets of $\{1,\ldots,N\}$ so as to minimize the RHS of (1.13). This idea leads to the definition of polar codes.

1.2.4 Polar codes

Given a B-DMC W, a G_N-coset code with parameter $(N,K,\mathscr{A},u_{\mathscr{A}^c})$ will be called a *polar code* for W if the information set $\mathscr{A}$ is chosen as a K-element subset of $\{1,\ldots,N\}$ such that $Z(W_N^{(i)}) \leq Z(W_N^{(j)})$ for all $i \in \mathscr{A}$, $j \in \mathscr{A}^c$.

Polar codes are channel-specific designs: a polar code for one channel may not be a polar code for another. The main result of this paper will be to show that polar coding achieves the symmetric capacity $I(W)$ of any given B-DMC W.

An alternative rule for polar code definition would be to specify $\mathscr{A}$ as a K-element subset of $\{1,\ldots,N\}$ such that $I(W_N^{(i)}) \geq I(W_N^{(j)})$ for all $i \in \mathscr{A}$, $j \in \mathscr{A}^c$. This alternative rule would also achieve $I(W)$. However, the rule based on the Bhattacharyya parameters has the advantage of being connected with an explicit bound on block error probability.

The polar code definition does not specify how the frozen vector $u_{\mathscr{A}^c}$ is to be chosen; it may be chosen at will. This degree of freedom in the choice of $u_{\mathscr{A}^c}$ simplifies the performance analysis of polar codes by allowing averaging over an ensemble. However, it is not for analytical convenience alone that we do not specify a precise rule for selecting $u_{\mathscr{A}^c}$, but also because it appears that the code performance is relatively insensitive to that choice. In fact, we prove in Sect. 4.6 that, for symmetric channels, any choice for $u_{\mathscr{A}^c}$ is as good as any other.

1.2.5 Coding theorems

Fix a B-DMC W and a number $R \geq 0$. Let $P_e(N,R)$ be defined as $P_e(N, \lfloor NR \rfloor, \mathscr{A})$ with $\mathscr{A}$ selected in accordance with the polar coding rule for W. Thus, $P_e(N,R)$ is the probability of block error under SC decoding for polar coding over W with block-length N and rate R, averaged over all choices for the frozen bits $u_{\mathscr{A}^c}$. The main coding result of this paper is the following:

Theorem 3 *For polar coding on a B-DMC W at any fixed rate $R < I(W)$, and any fixed $\beta < \frac{1}{2}$,*

$$P_e(N,R) = o(2^{-N^\beta}). \tag{1.15}$$

This theorem follows as an easy corollary to Theorem 2 and the bound (1.13), as we show in Sect. 4.3. For symmetric channels, we have the following stronger version of Theorem 3.

Theorem 4 *For any symmetric B-DMC W, any fixed $\beta < \frac{1}{2}$, and any fixed $R < I(W)$, consider any sequence of G_N-coset codes $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$ with N increasing to infinity, $K = \lfloor NR \rfloor$, $\mathscr{A}$ chosen in accordance with the polar coding rule for W, and $u_{\mathscr{A}^c}$ fixed arbitrarily. The block error probability under successive cancellation decoding satisfies*

$$P_e(N, K, \mathscr{A}, u_{\mathscr{A}^c}) = o(2^{-N^\beta}). \tag{1.16}$$

This is proved in Sect. 4.6. Note that for symmetric channels $I(W)$ equals the Shannon capacity of W.

1.2.6 A numerical example

The above results establish that polar codes achieve the symmetric capacity asymptotically. It is of interest to understand how quickly the polarization effect takes hold and what performance can be expected of polar codes under SC decoding in the non-asymptotic regime. To shed some light on this question, we give here a numerical example.

Let W be a BEC with erasure probability 1/2. For the BEC, there are exact formulas for computing the parameters $Z(W_N^{(i)})$, unlike other channels where this is a difficult problem. Figure 7 shows the rate vs. reliability trade-off for W using polar codes with block-lengths $N \in \{2^{10}, 2^{15}, 2^{20}\}$. This figure is obtained by using codes whose information sets are of the form $\mathscr{A}(\eta) \overset{\Delta}{=} \{i \in \{1,\dots,N\} : Z(W_N^{(i)}) < \eta\}$, where $0 \le \eta \le 1$ is a variable threshold parameter. There are two sets of three curves in the plot. The solid lines are plots of $R(\eta) \overset{\Delta}{=} |\mathscr{A}(\eta)|/N$ vs. $B(\eta) \overset{\Delta}{=} \sum_{i\in\mathscr{A}(\eta)} Z(W_N^{(i)})$. The dashed lines are plots of $R(\eta)$ vs. $L(\eta) \overset{\Delta}{=} \max_{i\in\mathscr{A}(\eta)}\{Z(W_N^{(i)})\}$. The parameter η is varied over a subset of $[0,1]$ to obtain the curves.

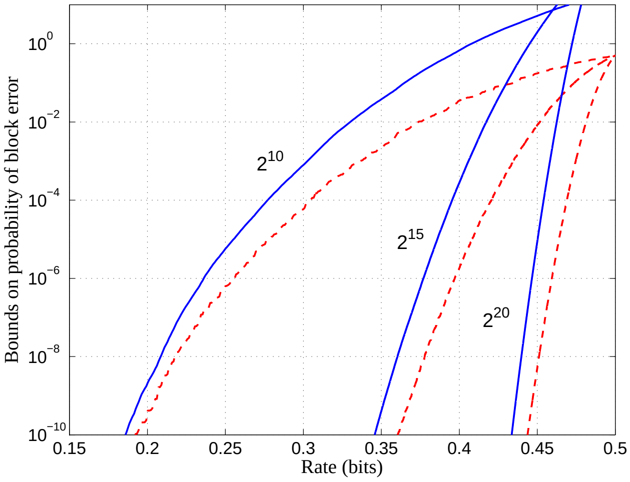

Fig. 1.5 Rate vs. reliability for polar coding and SC decoding at block-lengths 2^{10}, 2^{15}, and 2^{20} on a BEC with erasure probability $1/2$.

The parameter $R(\eta)$ corresponds to the code rate. The significance of $B(\eta)$ is also clear: it is an upper-bound on $P_e(\eta)$, the probability of block-error for polar coding at rate $R(\eta)$ under SC decoding. The parameter $L(\eta)$ is intended to serve as a lower bound to $P_e(\eta)$.

This example provides some empirical evidence that polar coding achieves channel capacity as the block-length is increased—a fact that will be established by exact proofs in the following. The example also shows that the rate of polarization is quite slow, limiting the practical impact of polar codes.

1.2.7 Complexity

An important issue about polar coding is the complexity of encoding, decoding, and code construction. The recursive structure of the channel polarization construction leads to low-complexity encoding and decoding algorithms for the class of G_N-coset codes, and in particular, for polar codes. The computational model we use in stating the following complexity results is a single CPU with a random access memory.

Theorem 5 *For the class of G_N-coset codes, the complexity of encoding and the complexity of successive cancellation decoding are both $O(N \log N)$ as functions of code block-length N.*

This theorem is proved in Sections 5.1 and 5.2. Notice that the complexity bounds in Theorem 5 are independent of the code rate and the way the frozen vector is chosen. The bounds hold even at rates above $I(W)$, but clearly this has no practical significance.

In general, no exact method is known for polar code construction that is of polynomial complexity. One exception is the case of a BEC for which we have a polar code construction algorithm with complexity $O(N)$. However, there exist approximation algorithms for constructing polar codes that have proven effective for practical purposes. These algorithms and their complexity will be discussed in Sect. 5.3.

1.3 Relations to Reed-Muller codes

Polar coding has much in common with Reed-Muller (RM) coding [11], [14]. According to one construction of RM codes, for any $N = 2^n$, $n \geq 0$, and $0 \leq K \leq N$, an RM code with block-length N and dimension K, denoted $\mathrm{RM}(N,K)$, is defined as a linear code whose generator matrix $G_{RM}(N,K)$ is obtained by deleting $(N-K)$ of the rows of $F^{\otimes n}$ so that none of the deleted rows has a larger Hamming weight (number of 1s in that row) than any of the remaining K rows. For instance,

$$G_{RM}(4,4) = F^{\otimes 2} = \begin{bmatrix} 1&0&0&0 \\ 1&1&0&0 \\ 1&0&1&0 \\ 1&1&1&1 \end{bmatrix}$$

and

$$G_{RM}(4,2) = \begin{bmatrix} 1&0&1&0 \\ 1&1&1&1 \end{bmatrix}.$$

This construction brings out the similarities between RM codes and polar codes. Since G_N and $F^{\otimes n}$ have the same set of rows for any $N = 2^n$, it is clear that RM codes belong to the class of G_N-coset codes. For example, $\mathrm{RM}(4,2)$ is the G_4-coset code with parameter $(4,2,\{2,4\},(0,0))$. So, RM coding and polar coding may be regarded as two alternative rules for selecting the information set $\mathscr{A}$ of a G_N-coset code of a given size (N,K). Unlike polar coding, RM coding selects the information set in a channel-independent manner; it is not as fine-tuned to the channel polarization phenomenon as polar coding is. It is shown in [1] that, at least for the class of BECs, the RM rule for information set selection leads to asymptotically unreliable codes under SC decoding. So, polar coding goes beyond RM coding in a non-trivial manner by paying closer attention to channel polarization. However, it is an open question whether RM codes fail to achieve channel capacity under ML decoding.

Another connection to existing work can be established by noting that polar codes are multi-level $|u|u+v|$ codes, which are a class of codes originating from Plotkin's method for code combining [13]. This connection is not surprising in view of the fact that RM codes are also multi-level $|u|u+v|$ codes [9, pp. 114-125]. However, unlike typical multi-level code constructions where one begins with specific small codes to build larger ones, in polar coding the multi-level code is obtained by expurgating rows of a full-order generator matrix, G_N, with respect to a channel-specific criterion. The special structure of G_N ensures that, no matter how expurgation is done, the resulting code is a multi-level $|u|u+v|$ code. In essence, polar coding enjoys the freedom to pick a multi-level code from an ensemble of such codes so as to suit the channel at hand, while conventional approaches to multi-level coding do not have this degree of flexibility.

1.4 Outline of the rest of notes

The rest of the notes is organized as follows. Chapter 2 examines the basic channel combining and splitting operation in detail, in particular, the recursive nature of that transform. In Chapter 3, we develop the main polarization result. In Chapter 4, we investigate the performance of polar codes and complete the proofs of polar coding theorems. Chapter 5 we discuss the complexity of the polar coding algorithms.

Chapter 2
Channel Transformation

Abstract This chapter describes the basic channel transformation operation and investigates the way $I(W)$ and $Z(W)$ get modified under this basic transformation. The basic transformation shows the first traces of polarization. The asymptotic analysis of polarization is left to the next chapter.

2.1 Recursive channel transformations

We have defined a blockwise channel combining and splitting operation by (1.2) and (1.3) which transformed N independent copies of W into $W_N^{(1)}, \dots, W_N^{(N)}$. The goal in this section is to show that this blockwise channel transformation can be broken recursively into single-step channel transformations.

We say that a pair of binary-input channels $W' : \mathscr{X} \to \tilde{\mathscr{Y}}$ and $W'' : \mathscr{X} \to \tilde{\mathscr{Y}} \times \mathscr{X}$ are obtained by a single-step transformation of two independent copies of a binary-input channel $W : \mathscr{X} \to \mathscr{Y}$ and write

$$(W,W) \mapsto (W',W'')$$

iff there exists a one-to-one mapping $f : \mathscr{Y}^2 \to \tilde{\mathscr{Y}}$ such that

$$W'(f(y_1,y_2)|u_1) = \sum_{u_2'} \frac{1}{2} W(y_1|u_1 \oplus u_2') W(y_2|u_2'), \tag{2.1}$$

$$W''(f(y_1,y_2),u_1|u_2) = \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2) \tag{2.2}$$

for all $u_1,u_2 \in \mathscr{X}$, $y_1,y_2 \in \mathscr{Y}$.

According to this, we can write $(W,W) \mapsto (W_2^{(1)},W_2^{(2)})$ for any given B-DMC W because

$$\begin{aligned} W_2^{(1)}(y_1^2|u_1) &\overset{\Delta}{=} \sum_{u_2} \frac{1}{2} W_2(y_1^2|u_1^2) \\ &= \sum_{u_2} \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2), \qquad (2.3) \\ W_2^{(2)}(y_1^2, u_1|u_2) &\overset{\Delta}{=} \frac{1}{2} W_2(y_1^2|u_1^2) \\ &= \frac{1}{2} W(y_1|u_1 \oplus u_2) W(y_2|u_2), \qquad (2.4) \end{aligned}$$

which are in the form of (2.1) and (2.2) by taking f as the identity mapping.

It turns out we can write, more generally,

$$(W_N^{(i)}, W_N^{(i)}) \mapsto (W_{2N}^{(2i-1)}, W_{2N}^{(2i)}). \qquad (2.5)$$

This follows as a corollary to the following:

Proposition 3 *For any $n \geq 0$, $N = 2^n$, $1 \leq i \leq N$,*

$$\begin{aligned} &W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) = \\ &\qquad \sum_{u_{2i}} \frac{1}{2}\, W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}) \qquad (2.6) \end{aligned}$$

and

$$\begin{aligned} &W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) = \\ &\qquad \frac{1}{2}\, W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}) W_N^{(i)}(y_{N+1}^{2N}, u_{1,e}^{2i-2}|u_{2i}). \qquad (2.7) \end{aligned}$$

This proposition is proved in the Appendix. The transform relationship (2.5) can now be justified by noting that (2.6) and (2.7) are identical in form to (2.1) and (2.2), respectively, after the following substitutions:

$$\begin{aligned} W &\leftarrow W_N^{(i)}, & W' &\leftarrow W_{2N}^{(2i-1)}, \\ W'' &\leftarrow W_{2N}^{(2i)}, & u_1 &\leftarrow u_{2i-1}, \\ u_2 &\leftarrow u_{2i}, & y_1 &\leftarrow (y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}), \\ y_2 &\leftarrow (y_{N+1}^{2N}, u_{1,e}^{2i-2}), & f(y_1, y_2) &\leftarrow (y_1^{2N}, u_1^{2i-2}). \end{aligned}$$

Thus, we have shown that the blockwise channel transformation from W^N to $(W_N^{(1)}, \ldots, W_N^{(N)})$ breaks at a local level into single-step channel transformations of the form (2.5). The full set of such transformations form a fabric as shown in Fig. 5 for $N = 8$. Reading from right to left, the figure starts with four copies of the transformation $(W, W) \mapsto (W_2^{(1)}, W_2^{(2)})$ and continues in *butterfly* patterns, each

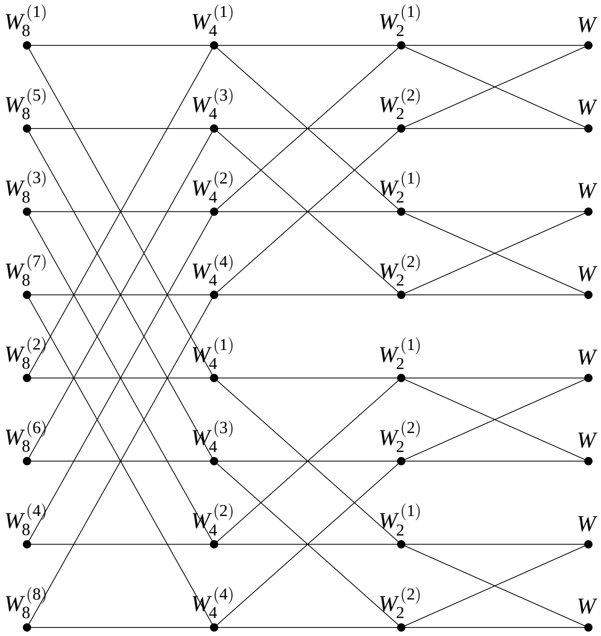
$W_8^{(1)}$ $W_4^{(1)}$ $W_2^{(1)}$ W
$W_8^{(5)}$ $W_4^{(3)}$ $W_2^{(2)}$ W
$W_8^{(3)}$ $W_4^{(2)}$ $W_2^{(1)}$ W
$W_8^{(7)}$ $W_4^{(4)}$ $W_2^{(2)}$ W
$W_8^{(2)}$ $W_4^{(1)}$ $W_2^{(1)}$ W
$W_8^{(6)}$ $W_4^{(3)}$ $W_2^{(2)}$ W
$W_8^{(4)}$ $W_4^{(2)}$ $W_2^{(1)}$ W
$W_8^{(8)}$ $W_4^{(4)}$ $W_2^{(2)}$ W

Fig. 2.1 The channel transformation process with $N = 8$ channels.

representing a channel transformation of the form $(W_{2^i}^{(j)}, W_{2^i}^{(j)}) \mapsto (W_{2^{i+1}}^{(2j-1)}, W_{2^{i+1}}^{(2j)})$. The two channels at the right end-points of the butterflies are always identical and independent. At the rightmost level there are 8 independent copies of W; at the next level to the left, there are 4 independent copies of $W_2^{(1)}$ and $W_2^{(2)}$ each; and so on. Each step to the left doubles the number of channel types, but halves the number of independent copies.

2.2 Transformation of rate and reliability

We now investigate how the rate and reliability parameters, $I(W_N^{(i)})$ and $Z(W_N^{(i)})$, change through a local (single-step) transformation (2.5). By understanding the local behavior, we will be able to reach conclusions about the overall transformation from W^N to $(W_N^{(1)}, \dots, W_N^{(N)})$. Proofs of the results in this section are given in the Appendix.

2.2.1 Local transformation of rate and reliability

Proposition 4 *Suppose $(W,W) \mapsto (W',W'')$ for some set of binary-input channels. Then,*

$$I(W') + I(W'') = 2I(W), \tag{2.8}$$
$$I(W') \leq I(W'') \tag{2.9}$$

with equality iff $I(W)$ equals 0 or 1.

The equality (2.8) indicates that the single-step channel transform preserves the symmetric capacity. The inequality (2.9) together with (2.8) implies that the symmetric capacity remains unchanged under a single-step transform, $I(W') = I(W'') = I(W)$, iff W is either a perfect channel or a completely noisy one. If W is neither perfect nor completely noisy, the single-step transform moves the symmetric capacity away from the center in the sense that $I(W') < I(W) < I(W'')$, thus helping polarization.

Proposition 5 *Suppose $(W,W) \mapsto (W',W'')$ for some set of binary-input channels. Then,*

$$Z(W'') = Z(W)^2, \tag{2.10}$$
$$Z(W') \leq 2Z(W) - Z(W)^2, \tag{2.11}$$
$$Z(W') \geq Z(W) \geq Z(W''). \tag{2.12}$$

Equality holds in (2.11) *iff W is a BEC. We have $Z(W') = Z(W'')$ iff $Z(W)$ equals 0 or 1, or equivalently, iff $I(W)$ equals 1 or 0.*

This result shows that reliability can only improve under a single-step channel transform in the sense that

$$Z(W') + Z(W'') \leq 2\,Z(W) \tag{2.13}$$

with equality iff W is a BEC.

Since the BEC plays a special role w.r.t. extremal behavior of reliability, it deserves special attention.

Proposition 6 *Consider the channel transformation $(W,W) \mapsto (W',W'')$. If W is a BEC with some erasure probability ε, then the channels W' and W'' are BECs with erasure probabilities $2\varepsilon - \varepsilon^2$ and ε^2, respectively. Conversely, if W' or W'' is a BEC, then W is BEC.*

2.2.2 Rate and reliability for $W_N^{(i)}$

We now return to the context at the end of Sect. 2.1.

Proposition 7 *For any B-DMC W, $N = 2^n$, $n \geq 0$, $1 \leq i \leq N$, the transformation $(W_N^{(i)}, W_N^{(i)}) \mapsto (W_{2N}^{(2i-1)}, W_{2N}^{(2i)})$ is rate-preserving and reliability-improving in the sense that*

$$I(W_{2N}^{(2i-1)}) + I(W_{2N}^{(2i)}) = 2\,I(W_N^{(i)}), \tag{2.14}$$

$$Z(W_{2N}^{(2i-1)}) + Z(W_{2N}^{(2i)}) \leq 2\,Z(W_N^{(i)}), \tag{2.15}$$

with equality in (2.15) *iff W is a BEC. Channel splitting moves the rate and reliability away from the center in the sense that*

$$I(W_{2N}^{(2i-1)}) \leq I(W_N^{(i)}) \leq I(W_{2N}^{(2i)}), \tag{2.16}$$

$$Z(W_{2N}^{(2i-1)}) \geq Z(W_N^{(i)}) \geq Z(W_{2N}^{(2i)}), \tag{2.17}$$

with equality in (2.16) *and* (2.17) *iff $I(W)$ equals 0 or 1. The reliability terms further satisfy*

$$Z(W_{2N}^{(2i-1)}) \leq 2Z(W_N^{(i)}) - Z(W_N^{(i)})^2, \tag{2.18}$$

$$Z(W_{2N}^{(2i)}) = Z(W_N^{(i)})^2, \tag{2.19}$$

$$Z(W_{2N}^{(2i)}) \leq Z(W_N^{(i)}) \leq Z(W_{2N}^{(2i-1)}), \tag{2.20}$$

with equality in (2.18) *iff W is a BEC and with equality on either side of* (2.20) *iff $I(W)$ is either 0 or 1. The cumulative rate and reliability satisfy*

$$\sum_{i=1}^{N} I(W_N^{(i)}) = NI(W), \tag{2.21}$$

$$\sum_{i=1}^{N} Z(W_N^{(i)}) \leq NZ(W), \tag{2.22}$$

with equality in (2.22) *iff W is a BEC.*

This result follows from Prop. 4 and Prop. 5 as a special case and no separate proof is needed. The cumulative relations (2.21) and (2.22) follow by repeated application of (2.14) and (2.15), respectively. The conditions for equality in Prop. 4 are stated in terms of W rather than $W_N^{(i)}$; this is possible because: (i) by Prop. 4, $I(W) \in \{0,1\}$ iff $I(W_N^{(i)}) \in \{0,1\}$; and (ii) W is a BEC iff $W_N^{(i)}$ is a BEC, which follows from Prop. 6 by induction.

For the special case that W is a BEC with an erasure probability ε, it follows from Prop. 4 and Prop. 6 that the parameters $\{Z(W_N^{(i)})\}$ can be computed through the recursion

$$\begin{aligned} Z(W_N^{(2j-1)}) &= 2\,Z(W_{N/2}^{(j)}) - Z(W_{N/2}^{(j)})^2, \\ Z(W_N^{(2j)}) &= Z(W_{N/2}^{(j)})^2, \end{aligned} \tag{2.23}$$

with $Z(W_1^{(1)}) = \varepsilon$. The parameter $Z(W_N^{(i)})$ equals the erasure probability of the channel $W_N^{(i)}$. The recursive relations (1.4) follow from (2.23) by the fact that $I(W_N^{(i)}) = 1 - Z(W_N^{(i)})$ for W a BEC.

Appendix

2.3 Proof of Proposition 3

To prove (2.6), we write

$$\begin{aligned}
W_{2N}^{(2i-1)}(y_1^{2N}, u_1^{2i-2}|u_{2i-1}) &= \sum_{u_{2i}^{2N}} \frac{1}{2^{2N-1}} W_{2N}(y_1^{2N}|u_1^{2N}) \\
&= \sum_{u_{2i,o}^{2N}, u_{2i,e}^{2N}} \frac{1}{2^{2N-1}} W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}) W_N(y_{N+1}^{2N}|u_{1,e}^{2N}) \\
&= \sum_{u_{2i}} \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N}|u_{1,e}^{2N}) \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}). \qquad (2.24)
\end{aligned}$$

By definition (1.3), the sum over $u_{2i+1,o}^{2N}$ for any fixed $u_{1,e}^{2N}$ equals

$$W_N^{(i)}(y_1^N, u_{1,o}^{2i-2} \oplus u_{1,e}^{2i-2}|u_{2i-1} \oplus u_{2i}),$$

because, as $u_{2i+1,o}^{2N}$ ranges over $\mathscr{X}^{N-i}$, $u_{2i+1,o}^{2N} \oplus u_{2i+1,e}^{2N}$ ranges also over $\mathscr{X}^{N-i}$. We now factor this term out of the middle sum in (2.24) and use (1.3) again to obtain (2.6). For the proof of (2.7), we write

$$\begin{aligned}
W_{2N}^{(2i)}(y_1^{2N}, u_1^{2i-1}|u_{2i}) &= \sum_{u_{2i+1}^{2N}} \frac{1}{2^{2N-1}} W_{2N}(y_1^{2N}|u_1^{2N}) \\
&= \frac{1}{2} \sum_{u_{2i+1,e}^{2N}} \frac{1}{2^{N-1}} W_N(y_{N+1}^{2N}|u_{1,e}^{2N}) \sum_{u_{2i+1,o}^{2N}} \frac{1}{2^{N-1}} W_N(y_1^N|u_{1,o}^{2N} \oplus u_{1,e}^{2N}).
\end{aligned}$$

By carrying out the inner and outer sums in the same manner as in the proof of (2.6), we obtain (2.7).

2.4 Proof of Proposition 4

Let us specify the channels as follows: $W : \mathscr{X} \to \mathscr{Y}$, $W' : \mathscr{X} \to \tilde{Y}$, and $W'' : \mathscr{X} \to \tilde{Y} \times \mathscr{X}$. By hypothesis there is a one-to-one function $f : \mathscr{Y} \to \tilde{\mathscr{Y}}$ such that (2.1) and (2.2) are satisfied. For the proof it is helpful to define an ensemble of RVs $(U_1,U_2,X_1,X_2,Y_1,Y_2,\tilde{Y})$ so that the pair (U_1,U_2) is uniformly distributed over $\mathscr{X}^2$, $(X_1,X_2) = (U_1 \oplus U_2,U_2)$, $P_{Y_1,Y_2|X_1,X_2}(y_1,y_2|x_1,x_2) = W(y_1|x_1)W(y_2|x_2)$, and $\tilde{Y} = f(Y_1,Y_2)$. We now have

$$\begin{aligned} W'(\tilde{y}|u_1) &= P_{\tilde{Y}|U_1}(\tilde{y}|u_1), \\ W''(\tilde{y},u_1|u_2) &= P_{\tilde{Y}U_1|U_2}(\tilde{y},u_1|u_2). \end{aligned}$$

From these and the fact that $(Y_1,Y_2) \mapsto \tilde{Y}$ is invertible, we get

$$\begin{aligned} I(W') &= I(U_1;\tilde{Y}) = I(U_1;Y_1Y_2), \\ I(W'') &= I(U_2;\tilde{Y}U_1) = I(U_2\,;Y_1Y_2U_1). \end{aligned}$$

Since U_1 and U_2 are independent, $I(U_2;Y_1Y_2U_1)$ equals $I(U_2;Y_1Y_2|U_1)$. So, by the chain rule, we have

$$I(W') + I(W'') = I(U_1U_2;Y_1Y_2) = I(X_1X_2;Y_1Y_2)$$

where the second equality is due to the one-to-one relationship between (X_1,X_2) and (U_1,U_2). The proof of (2.8) is completed by noting that $I(X_1X_2;Y_1Y_2)$ equals $I(X_1;Y_1) + I(X_2;Y_2)$ which in turn equals $2I(W)$.

To prove (2.9), we begin by noting that

$$\begin{aligned} I(W'') &= I(U_2;Y_1Y_2U_1) \\ &= I(U_2;Y_2) + I(U_2;Y_1U_1|Y_2) \\ &= I(W) + I(U_2;Y_1U_1|Y_2). \end{aligned}$$

This shows that $I(W'') \geq I(W)$. This and (2.8) give (2.9). The above proof shows that equality holds in (2.9) iff $I(U_2;Y_1U_1|Y_2) = 0$, which is equivalent to having

$$P_{U_1,U_2,Y_1|Y_2}(u_1,u_2,y_1|y_2) = P_{U_1,Y_1|Y_2}(u_1,y_1|y_2)P_{U_2|Y_2}(u_2|y_2)$$

for all (u_1,u_2,y_1,y_2) such that $P_{Y_2}(y_2) > 0$, or equivalently,

$$P_{Y_1,Y_2|U_1,U_2}(y_1,y_2|u_1,u_2)P_{Y_2}(y_2) = P_{Y_1,Y_2|U_1}(y_1,y_2|u_1)P_{Y_2|U_2}(y_2|u_2) \tag{2.25}$$

for all (u_1,u_2,y_1,y_2). Since $P_{Y_1,Y_2|U_1,U_2}(y_1,y_2|u_1,u_2) = W(y_1|u_1 \oplus u_2)W(y_2|u_2)$, eq. (2.25) can be written as

$$W(y_2|u_2)\left[W(y_1|u_1 \oplus u_2)P_{Y_2}(y_2) - P_{Y_1,Y_2}(y_1,y_2|u_1)\right] = 0. \tag{2.26}$$

Substituting $P_{Y_2}(y_2) = \frac{1}{2}W(y_2|u_2) + \frac{1}{2}W(y_2|u_2 \oplus 1)$ and

$$P_{Y_1,Y_2|U_1}(y_1,y_2|u_1) = \frac{1}{2}W(y_1|u_1 \oplus u_2)W(y_2|u_2) + \frac{1}{2}W(y_1|u_1 \oplus u_2 \oplus 1)W(y_2|u_2 \oplus 1)$$

into (2.26) and simplifying, we obtain

$$W(y_2|u_2)W(y_2|u_2\oplus 1)\,[W(y_1|u_1\oplus u_2) - W(y_1|u_1\oplus u_2\oplus 1)] = 0,$$

which for all four possible values of (u_1,u_2) is equivalent to

$$W(y_2|0)W(y_2|1)\;[W(y_1|0) - W(y_1|1)] = 0.$$

Thus, either there exists no y_2 such that $W(y_2|0)W(y_2|1) > 0$, in which case $I(W) = 1$, or for all y_1 we have $W(y_1|0) = W(y_1|1)$, which implies $I(W) = 0$.

2.5 Proof of Proposition 5

Proof of (2.10) is straightforward.

$$\begin{aligned}
Z(W'') &= \sum_{y_1^2,u_1} \sqrt{W''(f(y_1,y_2),u_1|0)}\;\sqrt{W''(f(y_1,y_2),u_1|1)}\\
&= \sum_{y_1^2,u_1} \frac{1}{2}\sqrt{W(y_1\mid u_1)W(y_2\mid 0)}\;\sqrt{W(y_1\mid u_1\oplus 1)W(y_2\mid 1)}\\
&= \sum_{y_2}\sqrt{W(y_2\mid 0)W(y_2\mid 1)}\;\sum_{u_1}\frac{1}{2}\;\sum_{y_1}\sqrt{W(y_1\mid u_1)W(y_1\mid u_1\oplus 1)}\\
&= Z(W)^2.
\end{aligned}$$

To prove (2.11), we put for shorthand $\alpha(y_1) = W(y_1|0)$, $\delta(y_1) = W(y_1|1)$, $\beta(y_2) = W(y_2|0)$, and $\gamma(y_2) = W(y_2|1)$, and write

$$\begin{aligned}
Z(W') &= \sum_{y_1^2}\sqrt{W'(f(y_1,y_2)|0)\,W'(f(y_1,y_2)|1)}\\
&= \sum_{y_1^2}\frac{1}{2}\sqrt{\alpha(y_1)\beta(y_2)+\delta(y_1)\gamma(y_2)}\;\sqrt{\alpha(y_1)\gamma(y_2)+\delta(y_1)\beta(y_2)}\\
&\le \sum_{y_1^2}\frac{1}{2}\left[\sqrt{\alpha(y_1)\beta(y_2)}+\sqrt{\delta(y_1)\gamma(y_2)}\right]\;\left[\sqrt{\alpha(y_1)\gamma(y_2)}+\sqrt{\delta(y_1)\beta(y_2)}\right]\\
&-\sum_{y_1^2}\sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)}
\end{aligned}$$

where the inequality follows from the identity

$$\begin{aligned}
&\left[\sqrt{(\alpha\beta+\delta\gamma)(\alpha\gamma+\delta\beta)}\right]^2 + 2\sqrt{\alpha\beta\delta\gamma}\,(\sqrt{\alpha}-\sqrt{\delta})^2(\sqrt{\beta}-\sqrt{\gamma})^2\\
&\qquad = \left[(\sqrt{\alpha\beta}+\sqrt{\delta\gamma})(\sqrt{\alpha\gamma}+\sqrt{\delta\beta}) - 2\sqrt{\alpha\beta\delta\gamma}\right]^2.
\end{aligned}$$

Next, we note that

$$\sum_{y_1^2} \alpha(y_1)\sqrt{\beta(y_2)\gamma(y_2)} = Z(W).$$

Likewise, each term obtained by expanding

$$(\sqrt{\alpha(y_1)\beta(y_2)} + \sqrt{\delta(y_1)\gamma(y_2)})(\sqrt{\alpha(y_1)\gamma(y_2)} + \sqrt{\delta(y_1)\beta(y_2)})$$

gives $Z(W)$ when summed over y_1^2. Also, $\sqrt{\alpha(y_1)\beta(y_2)\delta(y_1)\gamma(y_2)}$ summed over y_1^2 equals $Z(W)^2$. Combining these, we obtain the claim (2.11). Equality holds in (2.11) iff, for any choice of y_1^2, one of the following is true: $\alpha(y_1)\beta(y_2)\gamma(y_2)\delta(y_1) = 0$ or $\alpha(y_1) = \delta(y_1)$ or $\beta(y_2) = \gamma(y_2)$. This is satisfied if W is a BEC. Conversely, if we take $y_1 = y_2$, we see that for equality in (2.11), we must have, for any choice of y_1, either $\alpha(y_1)\delta(y_1) = 0$ or $\alpha(y_1) = \delta(y_1)$; this is equivalent to saying that W is a BEC.

To prove (2.12), we need the following result which states that the parameter $Z(W)$ is a convex function of the channel transition probabilities.

Lemma 3 *Given any collection of B-DMCs $W_j : \mathscr{X} \to \mathscr{Y}$, $j \in \mathscr{J}$, and a probability distribution Q on $\mathscr{J}$, define $W : \mathscr{X} \to \mathscr{Y}$ as the channel $W(y|x) = \sum_{j\in\mathscr{J}} Q(j)W_j(y|x)$. Then,*

$$\sum_{j\in\mathscr{J}} Q(j)Z(W_j) \le Z(W). \tag{2.27}$$

Proof. This follows by first rewriting $Z(W)$ in a different form and then applying Minkowsky's inequality [6, p. 524, ineq. (h)].

$$\begin{aligned} Z(W) &= \sum_y \sqrt{W(y|0)W(y|1)} \\ &= -1 + \frac{1}{2}\sum_y \left[\sum_x \sqrt{W(y|x)}\right]^2 \\ &\ge -1 + \frac{1}{2}\sum_y \sum_{j\in\mathscr{J}} Q(j) \left[\sum_x \sqrt{W_j(y|x)}\right]^2 \\ &= \sum_{j\in\mathscr{J}} Q(j)Z(W_j). \end{aligned}$$

We now write W' as the mixture

$$W'(f(y_1,y_2)|u_1) = \frac{1}{2}\left[W_0(y_1^2 \mid u_1) + W_1(y_1^2|u_1)\right]$$

where

$$W_0(y_1^2|u_1) = W(y_1|u_1)W(y_2|0),$$

$$W_1(y_1^2|u_1) = W(y_1|u_1 \oplus 1)W(y_2|1),$$

and apply Lemma 3 to obtain the claimed inequality

$$Z(W') \geq \frac{1}{2}[Z(W_0) + Z(W_1)] = Z(W).$$

Since $0 \leq Z(W) \leq 1$ and $Z(W'') = Z(W)^2$, we have $Z(W) \geq Z(W'')$, with equality iff $Z(W)$ equals 0 or 1. Since $Z(W') \geq Z(W)$, this also shows that $Z(W') = Z(W'')$ iff $Z(W)$ equals 0 or 1. So, by Prop. 1, $Z(W') = Z(W'')$ iff $I(W)$ equal to 1 or 0.

2.6 Proof of Proposition 6

From (2.1), we have the identities

$$\begin{aligned} &W'(f(y_1,y_2)|0)W'(f(y_1,y_2)|1) = \\ &\qquad \frac{1}{4}\left[W(y_1|0)^2 + W(y_1|1)^2\right]W(y_2|0)W(y_2|1) + \\ &\qquad \frac{1}{4}\left[W(y_2|0)^2 + W(y_2|1)^2\right]W(y_1|0)W(y_1|1) \end{aligned} \tag{2.28}$$

and

$$\begin{aligned} &W'(f(y_1,y_2)|0) - W'(f(y_1,y_2)|1) = \\ &\qquad \frac{1}{2}[W(y_1|0) - W(y_1|1)][W(y_2|0) - W(y_2|1)]. \end{aligned} \tag{2.29}$$

Suppose W is a BEC, but W' is not. Then, there exists (y_1,y_2) such that the left sides of (2.28) and (2.29) are both different from zero. From (2.29), we infer that neither y_1 nor y_2 is an erasure symbol for W. But then the RHS of (2.28) must be zero, which is a contradiction. Thus, W' must be a BEC. From (2.29), we conclude that $f(y_1,y_2)$ is an erasure symbol for W' iff either y_1 or y_2 is an erasure symbol for W. This shows that the erasure probability for W' is $2\varepsilon - \varepsilon^2$, where ε is the erasure probability of W.

Conversely, suppose W' is a BEC but W is not. Then, there exists y_1 such that $W(y_1|0)W(y_1|1) > 0$ and $W(y_1|0) - W(y_1|1) \neq 0$. By taking $y_2 = y_1$, we see that the RHSs of (2.28) and (2.29) can both be made non-zero, which contradicts the assumption that W' is a BEC.

The other claims follow from the identities

$$\begin{aligned} &W''(f(y_1,y_2),u_1|0)\,W''(f(y_1,y_2),u_1|1) \\ &\qquad = \frac{1}{4}W(y_1|u_1)W(y_1|u_1 \oplus 1)W(y_2|0)W(y_2|1) \end{aligned}$$

and

$$\begin{aligned}W''(f(y_1,y_2),u_1|0)&-W''(f(y_1,y_2),u_1|1)\\&=\frac{1}{2}\left[W(y_1|u_1)W(y_2|0)-W(y_1|u_1\oplus 1)W(y_2|1)\right].\end{aligned}$$

The arguments are similar to the ones already given and we omit the details, other than noting that $(f(y_1,y_2),u_1)$ is an erasure symbol for W'' iff both y_1 and y_2 are erasure symbols for W.

Chapter 3
Channel Polarization

Abstract This chapter proves the main polarization theorems.

3.1 Polarization Theorems

The goal of this chapter is to prove the main polarization theorems, restated below.

Theorem 1 *For any B-DMC W, the channels $\{W_N^{(i)}\}$ polarize in the sense that, for any fixed $\delta \in (0,1)$, as N goes to infinity through powers of two, the fraction of indices $i \in \{1,\ldots,N\}$ for which $I(W_N^{(i)}) \in (1-\delta, 1]$ goes to $I(W)$ and the fraction for which $I(W_N^{(i)}) \in [0,\delta)$ goes to $1-I(W)$.*

Theorem 2 *Let W be a B-DMC. For any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of sets $\{\mathscr{A}_N\}$ such that $\mathscr{A}_N \subset \{1,\ldots,N\}$, $|\mathscr{A}_N| \geq NR$, and*

$$\sum_{i \in \mathscr{A}_N} Z(W_N^{(i)}) = o(2^{-N^\beta}). \tag{3.1}$$

Conversely, if $R > 0$ and $\beta > \frac{1}{2}$, then for any sequence of sets $\{\mathscr{A}_N\}$ with $\mathscr{A}_N \subset \{1,\ldots,N\}$, $|\mathscr{A}_N| \geq NR$, we have

$$\max\{Z(W_N^{(i)}) : i \in \mathscr{A}_N\} = \omega(2^{-N^\beta}). \tag{3.2}$$

3.2 A stochastic process framework for analysis

The analysis is based on the recursive relationships depicted in Fig. 5; however, it will be more convenient to re-sketch Fig. 5 as a binary tree as shown in Fig. 6. The root node of the tree is associated with the channel W. The root W gives birth to an upper channel $W_2^{(1)}$ and a lower channel $W_2^{(2)}$, which are associated with the two nodes at level 1. The channel $W_2^{(1)}$ in turn gives birth to the channels $W_4^{(1)}$ and $W_4^{(2)}$, and so on. The channel $W_{2^n}^{(i)}$ is located at level n of the tree at node number i counting from the top.

There is a natural indexing of nodes of the tree in Fig. 6 by bit sequences. The root node is indexed with the null sequence. The upper node at level 1 is indexed with 0 and the lower node with 1. Given a node at level n with index $b_1b_2\cdots b_n$, the upper node emanating from it has the label $b_1b_2\cdots b_n0$ and the lower node $b_1b_2\cdots b_n1$. According to this labeling, the channel $W_{2^n}^{(i)}$ is situated at the node $b_1b_2\cdots b_n$ with $i = 1+\sum_{j=1}^n b_j 2^{n-j}$. We denote the channel $W_{2^n}^{(i)}$ located at node $b_1b_2\cdots b_n$ alternatively as $W_{b_1\ldots b_n}$.

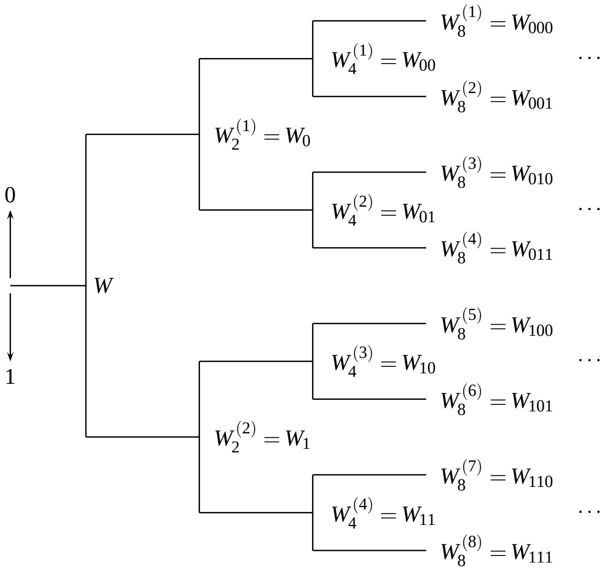

Fig. 3.1 The tree process for the recursive channel construction.

We define a random tree process, denoted $\{K_n; n \geq 0\}$, in connection with Fig. 6. The process begins at the root of the tree with $K_0 = W$. For any $n \geq 0$, given that $K_n = W_{b_1\cdots b_n}$, K_{n+1} equals $W_{b_1\cdots b_n0}$ or $W_{b_1\cdots b_n1}$ with probability 1/2 each. Thus, the path taken by $\{K_n\}$ through the channel tree may be thought of as being driven by a sequence of i.i.d. Bernoulli RVs $\{B_n; n = 1,2,\ldots\}$ where B_n equals 0 or 1 with equal probability. Given that $B_1,\ldots,B_n$ has taken on a sample value $b_1,\ldots,b_n$, the random channel process takes the value $K_n = W_{b_1\cdots b_n}$. In order to keep track of the rate and reliability parameters of the random sequence of channels K_n, we define the random processes $I_n = I(K_n)$ and $Z_n = Z(K_n)$.

For a more precise formulation of the problem, we consider the probability space $(\Omega, \mathscr{F}, P)$ where Ω is the space of all binary sequences $(b_1, b_2, \ldots) \in \{0,1\}^\infty$, $\mathscr{F}$ is the Borel field (BF) generated by the *cylinder sets* $S(b_1, \ldots, b_n) \overset{\Delta}{=} \{\omega \in \Omega : \omega_1 = b_1, \ldots, \omega_n = b_n\}$, $n \geq 1$, $b_1, \ldots, b_n \in \{0,1\}$, and P is the probability measure defined on $\mathscr{F}$ such that $P(S(b_1, \ldots, b_n)) = 1/2^n$. For each $n \geq 1$, we define $\mathscr{F}_n$ as the BF generated by the cylinder sets $S(b_1, \ldots, b_i)$, $1 \leq i \leq n$, $b_1, \ldots, b_i \in \{0,1\}$. We define $\mathscr{F}_0$ as the trivial BF consisting of the null set and Ω only. Clearly, $\mathscr{F}_0 \subset \mathscr{F}_1 \subset \cdots \subset \mathscr{F}$.

The random processes described above can now be formally defined as follows. For $\omega = (\omega_1, \omega_2, \ldots) \in \Omega$ and $n \geq 1$, define $B_n(\omega) = \omega_n$, $K_n(\omega) = W_{\omega_1 \cdots \omega_n}$, $I_n(\omega) = I(K_n(\omega))$, and $Z_n(\omega) = Z(K_n(\omega))$. For $n = 0$, define $K_0 = W$, $I_0 = I(W)$, $Z_0 = Z(W)$. It is clear that, for any fixed $n \geq 0$, the RVs B_n, K_n, I_n, and Z_n are measurable with respect to the BF $\mathscr{F}_n$.

3.3 Proof of Theorem 1

We will prove Theorem 1 by considering the stochastic convergence properties of the random sequences $\{I_n\}$ and $\{Z_n\}$.

Proposition 8 *The sequence of random variables and Borel fields $\{I_n, \mathscr{F}_n; n \geq 0\}$ is a martingale, i.e.,*

$$\mathscr{F}_n \subset \mathscr{F}_{n+1} \text{ and } I_n \text{ is } \mathscr{F}_n\text{-measurable}, \tag{3.3}$$

$$E[|I_n|] < \infty, \tag{3.4}$$

$$I_n = E[I_{n+1} | \mathscr{F}_n]. \tag{3.5}$$

Furthermore, the sequence $\{I_n; n \geq 0\}$ converges a.e. to a random variable I_∞ such that $E[I_\infty] = I_0$.

Proof. Condition (3.3) is true by construction and (3.4) by the fact that $0 \leq I_n \leq 1$. To prove (3.5), consider a cylinder set $S(b_1, \ldots, b_n) \in \mathscr{F}_n$ and use Prop. 7 to write

$$\begin{aligned} E[I_{n+1} | S(b_1, \cdots, b_n)] &= \frac{1}{2} I(W_{b_1 \cdots b_n 0}) + \frac{1}{2} I(W_{b_1 \cdots b_n 1}) \\ &= I(W_{b_1 \cdots b_n}). \end{aligned}$$

Since $I(W_{b_1 \cdots b_n})$ is the value of I_n on $S(b_1, \ldots, b_n)$, (3.5) follows. This completes the proof that $\{I_n, \mathscr{F}_n\}$ is a martingale. Since $\{I_n, \mathscr{F}_n\}$ is a uniformly integrable martingale, by general convergence results about such martingales (see, e.g., [3, Theorem 9.4.6]), the claim about I_∞ follows.

It should not be surprising that the limit RV I_∞ takes values a.e. in $\{0,1\}$, which is the set of fixed points of $I(W)$ under the transformation $(W, W) \mapsto (W_2^{(1)}, W_2^{(2)})$, as determined by the condition for equality in (2.9). For a rigorous proof of this statement, we take an indirect approach and bring the process $\{Z_n; n \geq 0\}$ also into the picture.

Proposition 9 *The sequence of random variables and Borel fields $\{Z_n, \mathscr{F}_n; n \geq 0\}$ is a supermartingale, i.e.,*

$$\mathscr{F}_n \subset \mathscr{F}_{n+1} \text{ and } Z_n \text{ is } \mathscr{F}_n\text{-measurable}, \tag{3.6}$$

$$E[|Z_n|] < \infty, \tag{3.7}$$

$$Z_n \geq E[Z_{n+1}|\mathscr{F}_n]. \tag{3.8}$$

Furthermore, the sequence $\{Z_n; n \geq 0\}$ converges a.e. to a random variable Z_∞ which takes values a.e. in $\{0,1\}$.

Proof. Conditions (3.6) and (3.7) are clearly satisfied. To verify (3.8), consider a cylinder set $S(b_1,\dots,b_n) \in \mathscr{F}_n$ and use Prop. 7 to write

$$\begin{aligned} E[Z_{n+1}|S(b_1,\dots,b_n)] &= \frac{1}{2}Z(W_{b_1\cdots b_n 0}) + \frac{1}{2}Z(W_{b_1\cdots b_n 1}) \\ &\leq Z(W_{b_1\cdots b_n}). \end{aligned}$$

Since $Z(W_{b_1\cdots b_n})$ is the value of Z_n on $S(b_1,\dots,b_n)$, (3.8) follows. This completes the proof that $\{Z_n, \mathscr{F}_n\}$ is a supermartingale. For the second claim, observe that the supermartingale $\{Z_n, \mathscr{F}_n\}$ is uniformly integrable; hence, it converges a.e. and in $\mathscr{L}^1$ to a RV Z_∞ such that $E[|Z_n - Z_\infty|] \to 0$ (see, e.g., [3, Theorem 9.4.5]). It follows that $E[|Z_{n+1} - Z_n|] \to 0$. But, by Prop. 7, $Z_{n+1} = Z_n^2$ with probability 1/2; hence, $E[|Z_{n+1} - Z_n|] \geq (1/2)E[Z_n(1-Z_n)] \geq 0$. Thus, $E[Z_n(1-Z_n)] \to 0$, which implies $E[Z_\infty(1-Z_\infty)] = 0$. This, in turn, means that Z_∞ equals 0 or 1 a.e.

Proposition 10 *The limit RV I_∞ takes values a.e. in the set $\{0,1\}$: $P(I_\infty = 1) = I_0$ and $P(I_\infty = 0) = 1 - I_0$.*

Proof. The fact that Z_∞ equals 0 or 1 a.e., combined with Prop. 1, implies that $I_\infty = 1 - Z_\infty$ a.e. Since $E[I_\infty] = I_0$, the rest of the claim follows.

As a corollary to Prop. 10, we can conclude that, as N tends to infinity, the symmetric capacity terms $\{I(W_N^{(i)} : 1 \leq i \leq N\}$ cluster around 0 and 1, except for a vanishing fraction. This completes the proof of Theorem 1.

3.4 Proof of the converse part of Theorem 2

We first prove the converse part of Theorem 2 which we restate as follows.

Proposition 11 *For any $\beta > 1/2$ and with $P(Z_0 > 0) > 0$,*

$$\lim_{n\to\infty} P\big(Z_n < 2^{-2^{n\beta}}\big) = 0. \tag{3.9}$$

Proof. Observe that the random process Z_n is lower-bounded by the process $\{L_n : n \in \mathbb{N}\}$ defined by $L_0 := Z_0$ and for $n \geq 1$

$$\begin{aligned} L_n &= L_{n-1}^2 && \text{when } B_n = 1, \\ L_n &= L_{n-1} && \text{when } B_n = 0. \end{aligned}$$

Thus, $L_n = L_0^{2^{S_n}}$ where $S_n := \sum_{i=1}^n B_i$. So, we have

$$\begin{aligned} P\big(Z_n \leq 2^{-2^{\beta n}}\big) &\leq P\big(L_n \leq 2^{-2^{\beta n}}\big) \\ &= P\bigg(S_n \geq n\beta - \log_2(-\log_2(Z_0))\bigg). \end{aligned}$$

For $\beta > \frac{1}{2}$, this last probability goes to zero as n increases by the law of large numbers.

3.5 Proof of Theorem 2: The direct part

In this part, we will establish the direct part of Theorem 2 which may be stated as follows.

Proposition 12 *For any given $\beta < \frac{1}{2}$ and $\varepsilon > 0$, there exists n such that*

$$P\big(Z_n < 2^{-2^{n\beta}}\big) \geq I_0 - \varepsilon. \tag{3.10}$$

The proof of this result is quite lengthy and will be split into several parts. It will be convenient to introduce some notation and state an elementary fact before beginning the proof.

For $n > m \geq 0$ and $0 \leq \beta \leq 1$, define $S_{m,n} = \sum_{i=m+1}^n B_i$ and

$$\mathscr{S}_{m,n}(\beta) = \{\omega \in \Omega : \ S_{m,n}(\omega) > (n-m)\beta\}.$$

By Chernoff's bound (see, e.g., [6, p. 531]), for $0 \leq \beta \leq \frac{1}{2}$, the probability of this set is bounded as

$$P\left[\mathscr{S}_{m,n}(\beta)\right] \geq 1 - 2^{-(n-m)[1-\mathscr{H}(\beta)]} \tag{3.11}$$

where $\mathscr{H}(\beta) = -\beta\log_2(\beta) - (1-\beta)\log_2(1-\beta)$ is the binary entropy function. Clearly, for $0 \leq \beta < 1/2$, the probability of $\mathscr{S}_{m,n}$ goes to 1 as $(n-m)$ increases. Define $n_0(\beta, \varepsilon)$ as the smallest value of $(n-m)$ such that the RHS of (3.11) is greater than or equal to $1 - \varepsilon$.

3.5.1 A bootstrapping method

We first give a bound to majorize the process $\{Z_n\}$ on a sample function basis. For this it is more convenient to consider the logarithmic process $V_n := \log_2(Z_n)$. This process evolves as

$$\begin{aligned} V_{i+1} &= 2V_i && \text{when } B_{i+1} = 1, \\ V_{i+1} &\le V_i + 1 && \text{when } B_{i+1} = 0. \end{aligned}$$

Thus, at each step either the value is doubled or incremented by an amount not exceeding one. In terms of this process, we wish to show that with probability close to I_0 we have $V_n \approx -2^{\frac{n}{2}}$.

The following lemma is key to analyzing the behavior of the process $\{V_n\}$.

Lemma 4 *Let $A : \mathbb{R} \to \mathbb{R}$, $A(x) = x + 1$ denote adding one, and $D : \mathbb{R} \to \mathbb{R}$, $D(x) = 2x$ denote doubling. Suppose a sequence of numbers $a_0, a_1, \ldots, a_n$ is defined by specifying a_0 and the recursion*

$$a_{i+1} = f_i(a_i)$$

with $f_i \in \{A, D\}$. Suppose $\big|\{0 \le i \le n-1 : f_i = D\}\big| = k$ and $\big|\{0 \le i \le n-1 : f_i = A\}\big| = n-k$, i.e., during the first n iterations of the recursion we encounter doubling k times and adding-one n − k times. Then

$$a_n \le D^{(k)}\big(A^{(n-k)}(a_0)\big) = 2^k(a_0 + n - k).$$

Proof. Observe that the upper bound on a_n corresponds to choosing

$$f_0 = \cdots f_{n-k-1} = A \quad \text{and} \quad f_{n-k} = \cdots = f_{n-1} = D.$$

We will show that any other choice of $\{f_i\}$ can be modified to yield a higher value of a_n. To that end suppose $\{f_i\}$ is not chosen as above. Then there exists $j \in \{1, \ldots, n-1\}$ for which $f_{j-1} = D$ and $f_j = A$. Define $\{f_i'\}$ by swapping f_j and f_{j-1}, i.e.,

$$f_i' = \begin{cases} A & i = j-1 \\ D & i = j \\ f_i & \text{else} \end{cases}$$

and let $\{a_i'\}$ denote the sequence that results from $\{f_i'\}$. Then

$$\begin{aligned} a_i' &= a_i \quad \text{for } i < j \\ a_j' &= a_{j-1} + 1 \\ a_{j+1}' &= 2a_j' = 2a_{j-1} + 2 \\ &> 2a_{j-1} + 1 = a_{j+1}. \end{aligned}$$

Since the recursion from $j+1$ onwards is identical for the $\{f_i\}$ and $\{f_i'\}$ sequences, and since both A and D are order preserving, $a'_{j+1} > a_{j+1}$ implies that $a'_n > a_n$.

By Lemma 4, we can write for any $n > m$

$$\begin{aligned} V_n &\leq \left[V_m + (n-m) - S_{m,n}\right] 2^{S_{m,n}} \\ &\leq \left[V_m + (n-m)\right] 2^{S_{m,n}} \end{aligned}$$

The process $\{V_n\}$ takes values in $(-\infty, 0]$ and the above bound is effective only when $V_m + (n-m)$ is less than 0. This means that for fixed m, there is a limit to how large n can be taken before rendering the bound useless. On the other hand, in order to obtain the desired rate of exponential convergence one wishes to take n much larger than m so that the exponent can be approximated with high probability as

$$S_{m,n} \approx n/2.$$

Fortunately, by applying the same bound repeatedly these two conflicting constraints on the choice of n can be alleviated. For example, applying the bound first over $[m,k]$ and then over $[k,n]$ we obtain

$$V_n \leq \left[\left(V_m + (k-m)\right) 2^{S_{m,k}} + (n-k)\right] 2^{S_{n,k}} \tag{3.12}$$

Now, a value of k modestly larger than m can ensure that V_k takes on a sufficiently large negative value to ensure that we can choose $n \gg k$. This will be shown below. However, still one needs to be able to begin with a large enough negative value for V_m to initiate the bootstrapping operation. The following result states that this can be done.

Proposition 13 *For any given $\varepsilon > 0$ and there exists $m_0(\varepsilon)$ such that for all $m \geq m_0(\varepsilon)$*

$$P\left(V_m \leq -2m\right) \geq I_0 - \varepsilon \tag{3.13}$$

Accepting the validity of Proposition 13 momentarily, we will show how to complete the proof of Proposition 12. We will prove Proposition 13 in the following two subsections.

Let $m \geq m_0(\varepsilon/3)$ be arbitrary. Set $k = 2m$ and $n = m^2$. Then, with probability at least $I_0 - \varepsilon/3$, we have by (3.12) that

$$V_{m^2} \leq \left(-m2^{S_{m,2m}} + (m^2 - 2m)\right) 2^{S_{2m,m^2}}$$

For any given $\beta < 1/2$, we can choose $\beta' \in (\beta, 1/2)$ such that for m sufficiently large we have

$$P\left(S_{m,2m} > \beta' m\right) \geq 1 - \varepsilon/3$$

and

$$P\left(S_{2m,m^2} > \beta'(m^2 - m)\right) \geq 1 - \varepsilon/3$$

So, for such m we have with probability at least $I_0 - \varepsilon$

$$V_{m^2} \le \left[-m2^{m\beta'} + (m^2 - 2m)\right] 2^{(m^2-2m)\beta'}.$$

For a non-trivial bound we need to ensure that the term in square brackets is bounded away from zero on the negative side. So, we impose the following additional constraint on m:

$$\left[-m2^{m\beta'} + (m^2 - 2m)\right] < -1$$

which clearly can be met by choosing m large enough. Then, for all m satisfying all the constraints above we have

$$V_{m^2} \le -2^{(m^2-2m)\beta'}$$

with probability at least $I_0 - \varepsilon$. This, written in terms of $n = m^2$ reads as

$$V_n \le -2^{(n-o(n))\beta'} \le -2^{n\beta}$$

where the second inequality holds for n large enough since $\beta' > \beta$.

3.5.2 Sealing the process in $[0, \zeta]$

The proof of Proposition 13 also contains a bootstrapping argument, but of a different type. We first establish a result that “seals” as much of the sample paths of $\{Z_n\}$ as possible in a small interval around zero. For $\zeta \ge 0$ and $\ell \ge 0$, define

$$\mathscr{T}_\ell(\zeta) \stackrel{\Delta}{=} \{\omega \in \Omega : Z_i(\omega) \le \zeta \text{ for all } i \ge \ell\}.$$

Lemma 5 *For any $\zeta > 0$ and $\varepsilon > 0$, there exists $\ell_0(\zeta, \varepsilon)$ such that for all $\ell \ge \ell_0$*

$$P\left[\mathscr{T}_\ell(\zeta)\right] \ge I_0 - \varepsilon.$$

Proof. Fix $\zeta > 0$. Let $\Omega_0 \stackrel{\Delta}{=} \{\omega \in \Omega : \lim_{n\to\infty} Z_n(\omega) = 0\}$. By Prop. 10, $P(\Omega_0) = I_0$. Fix $\omega \in \Omega_0$. $Z_n(\omega) \to 0$ implies that there exists $n_0(\omega, \zeta)$ such that $n \ge n_0(\omega, \zeta) \Rightarrow Z_n(\omega) \le \zeta$. Thus, $\omega \in \mathscr{T}_\ell(\zeta)$ for some m. So, $\Omega_0 \subset \bigcup_{\ell=1}^{\infty} \mathscr{T}_\ell(\zeta)$. Therefore, $P(\bigcup_{\ell=1}^{\infty} \mathscr{T}_\ell(\zeta)) \ge P(\Omega_0)$. Since $\mathscr{T}_\ell(\zeta) \uparrow \bigcup_{\ell=1}^{\infty} \mathscr{T}_\ell(\zeta)$, by the monotone convergence property of a measure, $\lim_{\ell\to\infty} P\left[\mathscr{T}_\ell(\zeta)\right] = P\left[\bigcup_{\ell=1}^{\infty} \mathscr{T}_\ell(\zeta)\right]$. So, $\lim_{\ell\to\infty} P\left[\mathscr{T}_\ell(\zeta)\right] \ge I_0$. It follows that, for any $\zeta > 0$, $\varepsilon > 0$, there exists a finite $\ell_0 = \ell_0(\zeta, \varepsilon)$ such that, for all $\ell \ge \ell_0$, $P\left[\mathscr{T}_\ell(\zeta)\right] \ge I_0 - \varepsilon$. This completes the proof.

3.5.3 Proof of Proposition 13

For $\omega \in \mathscr{T}_\ell(\zeta)$ and $i \geq \ell$, we have

$$\frac{Z_{i+1}(\omega)}{Z_i(\omega)} \leq \begin{cases} 2, & \text{if } B_{i+1}(\omega) = 0 \\ \zeta, & \text{if } B_{i+1}(\omega) = 1 \end{cases}$$

which implies

$$Z_m(\omega) \leq Z_\ell(\omega)\, 2^{m-\ell-S_{\ell,m}(\omega)}\, \zeta^{S_{\ell,m}(\omega)}, \quad \omega \in \mathscr{T}_\ell(\zeta),\ m > \ell.$$

This gives

$$Z_m(\omega) \leq Z_\ell(\omega) \left(2^{1-\beta}\, \zeta^\beta\right)^{m-\ell}, \quad \omega \in \mathscr{T}_\ell(\zeta) \cap \mathscr{S}_{\ell,m}(\beta).$$

Now, we set $\zeta = \zeta_0 := 2^{-9}$, $\beta = \beta_0 := 9/20$, $m = (7\ell/3)$, and note that $Z_\ell \leq 1$, to obtain

$$Z_m(\omega) \leq 2^{-2m}, \quad \omega \in \mathscr{T}_{(3m/7)}(\zeta_0) \cap \mathscr{S}_{(3m/7),m}(\beta_0). \tag{3.14}$$

The bound (3.11) and Lemma 5 ensure that there exists $m_0(\varepsilon)$ such that, for all $m \geq m_0(\varepsilon)$, (3.14) holds with probability greater than $I_0 - \varepsilon$. Specifically, it suffices to take m greater than both $(7/4)n_0(\beta_0, \varepsilon/2)$ and $(7/3)\ell_0(\zeta_0, \varepsilon/2)$.

3.5.4 Complementary remarks

Theorem 2 was first proved in [2] and the proof of the theorem proved above followed that paper closely. The channel polarization result as expressed by Theorem 2 does not show an explicit dependence on the rate parameter R except for the condition that $R < I_0$. Rate-dependent refinements of this theorem have appeared in [18], [8], [17] soon after the publication of [2]. For a more recent work on the same subject, see [7]. To state this refined polarization theorem, let $Q : \mathbb{R} \to [0,1]$ denote the complementary cumulative distribution function for the standard normal distribution:

$$Q(t) = \frac{1}{\sqrt{2\pi}} \int_t^\infty e^{-u^2/2} du.$$

Let Q^{-1} denote the inverse of Q. Then, the refined result can be stated in the present notation as follows.

Theorem 6 *For any $0 \leq R < I(W)$, the Bhattacharyya random process in polarization has asymptotic probabilities given by*

$$P\left(Z_n \leq 2^{-2^{[n+Q^{-1}(R/I_0)\sqrt{n}]/2+o(\sqrt{n})}}\right) \to R.$$

3.6 A side result

It is interesting that Propositon 9 gives a new interpretation to the symmetric capacity $I(W)$ as the probability that the random process $\{Z_n; n \geq 0\}$ converges to zero. Here, we use this to strengthen the lower bound in (0.1).

Proposition 14 *For any B-DMC W, we have $I(W) + Z(W) \geq 1$ with equality iff W is a BEC.*

This result can be interpreted as saying that, among all B-DMCs W, the BEC presents the most favorable rate-reliability trade-off: it minimizes $Z(W)$ (maximizes reliability) among all channels with a given symmetric capacity $I(W)$; equivalently, it minimizes $I(W)$ required to achieve a given level of reliability $Z(W)$.

Proof. Consider two channels W and W' with $Z(W) = Z(W') \overset{\Delta}{=} z_0$. Suppose that W' is a BEC. Then, W' has erasure probability z_0 and $I(W') = 1 - z_0$. Consider the random processes $\{Z_n\}$ and $\{Z'_n\}$. By the condition for equality in (2.18), the process $\{Z_n\}$ is stochastically dominated by $\{Z'_n\}$ in the sense that $P(Z_n \leq z) \geq P(Z'_n \leq z)$ for all $n \geq 1$, $0 \leq z \leq 1$. Thus, the probability of $\{Z_n\}$ converging to zero is lower-bounded by the probability that $\{Z'_n\}$ converges to zero, i.e., $I(W) \geq I(W')$. This implies $I(W) + Z(W) \geq 1$.

Chapter 4
Polar Coding

Abstract We show in this section that polar coding can achieve the symmetric capacity $I(W)$ of any B-DMC W.

4.1 Plan of chapter

The main technical task in this chapter will be to prove Prop. 2. We will carry out the analysis over the class of G_N-coset codes before specializing the discussion to polar codes. Recall that individual G_N-coset codes are identified by a parameter vector $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$. In the analysis, we will fix the parameters $(N, K, \mathscr{A})$ while keeping $u_{\mathscr{A}^c}$ free to take any value over $\mathscr{X}^{N-K}$. In other words, the analysis will be over the ensemble of 2^{N-K} G_N-coset codes with a fixed $(N, K, \mathscr{A})$. The decoder in the system will be the SC decoder described in Sect. 1.2.2.

4.2 A probabilistic setting for the analysis

Let $(\mathscr{X}^N \times \mathscr{Y}^N, P)$ be a probability space with the probability assignment

$$P(\{(u_1^N, y_1^N)\}) \overset{\Delta}{=} 2^{-N} W_N(y_1^N | u_1^N) \tag{4.1}$$

for all $(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N$. On this probability space, we define an ensemble of random vectors $(U_1^N, X_1^N, Y_1^N, \hat{U}_1^N)$ that represent, respectively, the input to the synthetic channel W_N, the input to the product-form channel W^N, the output of W^N (and also of W_N), and the decisions by the decoder. For each sample point $(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N$, the first three vectors take on the values $U_1^N(u_1^N, y_1^N) = u_1^N$, $X_1^N(u_1^N, y_1^N) = u_1^N G_N$, and $Y_1^N(u_1^N, y_1^N) = y_1^N$, while the decoder output takes on the value $\hat{U}_1^N(u_1^N, y_1^N)$ whose coordinates are defined recursively as

$$\hat{U}_i(u_1^N, y_1^N) = \begin{cases} u_i, & i \in \mathscr{A}^c \\ h_i(y_1^N, \hat{U}_1^{i-1}(u_1^N, y_1^N)), & i \in \mathscr{A} \end{cases} \tag{4.2}$$

for $i = 1, \dots, N$.

A realization $u_1^N \in \mathscr{X}^N$ for the input random vector U_1^N corresponds to sending the data vector $u_{\mathscr{A}}$ together with the frozen vector $u_{\mathscr{A}^c}$. As random vectors, the data part $U_{\mathscr{A}}$ and the frozen part $U_{\mathscr{A}^c}$ are uniformly distributed over their respective ranges and statistically independent. By treating $U_{\mathscr{A}^c}$ as a random vector over $\mathscr{X}^{N-K}$, we obtain a convenient method for analyzing code performance averaged over all codes in the ensemble $(N, K, \mathscr{A})$.

The main event of interest in the following analysis is the block error event under SC decoding, defined as

$$\mathscr{E} \overset{\Delta}{=} \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : \hat{U}_{\mathscr{A}}(u_1^N, y_1^N) \neq u_{\mathscr{A}}\}. \tag{4.3}$$

Since the decoder never makes an error on the frozen part of U_1^N, i.e., $\hat{U}_{\mathscr{A}^c}$ equals $U_{\mathscr{A}^c}$ with probability one, that part has been excluded from the definition of the block error event.

The probability of error terms $P_e(N, K, \mathscr{A})$ and $P_e(N, K, \mathscr{A}, u_{\mathscr{A}^c})$ that were defined in Sect. 1.2.3 can be expressed in this probability space as

$$\begin{aligned} P_e(N, K, \mathscr{A}) &= P(\mathscr{E}), \\ P_e(N, K, \mathscr{A}, u_{\mathscr{A}^c}) &= P(\mathscr{E} \mid \{U_{\mathscr{A}^c} = u_{\mathscr{A}^c}\}), \end{aligned} \tag{4.4}$$

where $\{U_{\mathscr{A}^c} = u_{\mathscr{A}^c}\}$ denotes the event $\{(\tilde{u}_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : \tilde{u}_{\mathscr{A}^c} = u_{\mathscr{A}^c}\}$.

4.3 Proof of Proposition 2

We may express the block error event as $\mathscr{E} = \cup_{i \in \mathscr{A}} \mathscr{B}_i$ where

$$\mathscr{B}_i \overset{\Delta}{=} \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}(u_1^N, y_1^N),\ u_i \neq \hat{U}_i(u_1^N, y_1^N)\} \tag{4.5}$$

is the event that the first decision error in SC decoding occurs at stage i. We notice that

$$\begin{aligned} \mathscr{B}_i &= \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}(u_1^N, y_1^N), u_i \neq h_i(y_1^N, \hat{U}_1^{i-1}(u_1^N, y_1^N))\} \\ &= \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : u_1^{i-1} = \hat{U}_1^{i-1}(u_1^N, y_1^N), u_i \neq h_i(y_1^N, u_1^{i-1})\} \\ &\subset \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : u_i \neq h_i(y_1^N, u_1^{i-1})\} \\ &\subset \mathscr{E}_i \end{aligned}$$

where

$$\mathscr{E}_i \overset{\Delta}{=} \{(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N : W_N^{(i-1)}(y_1^N, u_1^{i-1} \mid u_i) \leq W_N^{(i-1)}(y_1^N, u_1^{i-1} \mid u_i \oplus 1)\}. \tag{4.6}$$

Thus, we have

$$\mathscr{E} \subset \bigcup_{i \in \mathscr{A}} \mathscr{E}_i, \qquad P(\mathscr{E}) \leq \sum_{i \in \mathscr{A}} P(\mathscr{E}_i).$$

For an upper bound on $P(\mathscr{E}_i)$, note that

$$\begin{aligned} P(\mathscr{E}_i) &= \sum_{u_1^N, y_1^N} \frac{1}{2^N} W_N(y_1^N \mid u_1^N) 1_{\mathscr{E}_i}(u_1^N, y_1^N) \\ &\leq \sum_{u_1^N, y_1^N} \frac{1}{2^N} W_N(y_1^N \mid u_1^N) \sqrt{\frac{W_N^{(i)}(y_1^N, u_1^{i-1} | u_i \oplus 1)}{W_N^{(i)}(y_1^N, u_1^{i-1} | u_i)}} \\ &= Z(W_N^{(i)}). \end{aligned} \tag{4.7}$$

We conclude that

$$P(\mathscr{E}) \leq \sum_{i \in \mathscr{A}} Z(W_N^{(i)}),$$

which is equivalent to (1.13). This completes the proof of Prop. 2. The main coding theorem of the paper now follows readily.

4.4 Proof of Theorem 3

By Theorem 2, for any fixed rate $R < I(W)$ and constant $\beta < \frac{1}{2}$, there exists a sequence of sets $\{\mathscr{A}_N\}$ such that $\mathscr{A}_N \subset \{1, \ldots, N\}$, $|\mathscr{A}_N| \geq NR$, and

$$\sum_{i \in \mathscr{A}_N} Z(W_N^{(i)}) = o(2^{-N^\beta}). \tag{4.8}$$

In particular, the bound (4.8) holds if $\mathscr{A}_N$ is chosen in accordance with the polar coding rule because by definition this rule minimizes the sum in (4.8). Combining this fact about the polar coding rule with Prop. 2, Theorem 3 follows.

4.5 Symmetry under channel combining and splitting

Let $W : \mathscr{X} \to \mathscr{Y}$ be a symmetric B-DMC with $\mathscr{X} = \{0, 1\}$ and $\mathscr{Y}$ arbitrary. By definition, there exists a a permutation π_1 on $\mathscr{Y}$ such that (i) $\pi_1^{-1} = \pi_1$ and (ii) $W(y|1) = W(\pi_1(y)|0)$ for all $y \in \mathscr{Y}$. Let π_0 be the identity permutation on $\mathscr{Y}$.

Clearly, the permutations (π_0, π_1) form an abelian group under function composition. For a compact notation, we will write $x \cdot y$ to denote $\pi_x(y)$, for $x \in \mathscr{X}, y \in \mathscr{Y}$.

Observe that $W(y|x \oplus a) = W(a \cdot y|x)$ for all $a, x \in \mathscr{X}, y \in \mathscr{Y}$. This can be verified by exhaustive study of possible cases or by noting that $W(y|x \oplus a) = W((x \oplus a) \cdot y|0) = W(x \cdot (a \cdot y)|0) = W(a \cdot y|x)$. Also observe that $W(y|x \oplus a) = W(x \cdot y|a)$ as $\oplus$ is a commutative operation on $\mathscr{X}$.

For $x_1^N \in \mathscr{X}^N$, $y_1^N \in \mathscr{Y}^N$, let

$$x_1^N \cdot y_1^N \overset{\Delta}{=} (x_1 \cdot y_1, \dots, x_N \cdot y_N). \tag{4.9}$$

This associates to each element of $\mathscr{X}^N$ a permutation on $\mathscr{Y}^N$.

Proposition 15 *If a B-DMC W is symmetric, then W^N is also symmetric in the sense that*

$$W^N(y_1^N|x_1^N \oplus a_1^N) = W^N(x_1^N \cdot y_1^N|a_1^N) \tag{4.10}$$

for all $x_1^N, a_1^N \in \mathscr{X}^N$, $y_1^N \in \mathscr{Y}^N$.

The proof is immediate and omitted.

Proposition 16 *If a B-DMC W is symmetric, then the channels W_N and $W_N^{(i)}$ are also symmetric in the sense that*

$$W_N(y_1^N \mid u_1^N) = W_N(a_1^N G_N \cdot y_1^N \mid u_1^N \oplus a_1^N), \tag{4.11}$$

$$W_N^{(i)}(y_1^N, u_1^{i-1} \mid u_i) = W_N^{(i)}(a_1^N G_N \cdot y_1^N, u_1^{i-1} \oplus a_1^{i-1} \mid u_i \oplus a_i) \tag{4.12}$$

for all $u_1^N, a_1^N \in \mathscr{X}^N$, $y_1^N \in \mathscr{Y}^N$, $N = 2^n$, $n \geq 0$, $1 \leq i \leq N$.

Proof. Let $x_1^N = u_1^N G_N$ and observe that $W_N(y_1^N \mid u_1^N) = \prod_{i=1}^N W(y_i \mid x_i) = \prod_{i=1}^N W(x_i \cdot y_i \mid 0) = W_N(x_1^N \cdot y_1^N \mid 0_1^N)$. Now, let $b_1^N = a_1^N G_N$, and use the same reasoning to see that $W_N(b_1^N \cdot y_1^N \mid u_1^N \oplus a_1^N) = W_N((x_1^N \oplus b_1^N) \cdot (b_1^N \cdot y_1^N) \mid 0_1^N) = W_N(x_1^N \cdot y_1^N \mid 0_1^N)$. This proves the first claim. To prove the second claim, we use the first result.

$$\begin{aligned} W_N^{(i)}(y_1^N, u_1^{i-1} \mid u_i) &= \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N(y_1^N \mid u_1^N) \\ &= \sum_{u_{i+1}^N} \frac{1}{2^{N-1}} W_N(a_1^N G_N \cdot y_1^N \mid u_1^N \oplus a_1^N) \\ &= W_N(a_1^N G_N \cdot y_1^N, u_1^{i-1} \oplus a_1^{i-1} \mid u_i \oplus a_i) \end{aligned}$$

where we used the fact that the sum over $u_{i+1}^N \in \mathscr{X}^{N-i}$ can be replaced with a sum over $u_{i+1}^N \oplus a_{i+1}^N$ for any fixed a_1^N since $\{u_{i+1}^N \oplus a_{i+1}^N : u_{i+1}^N \in \mathscr{X}^{N-i}\} = X^{N-i}$.

4.6 Proof of Theorem 4

We return to the analysis in Sect. 4.3 and consider a code ensemble $(N, K, \mathscr{A})$ under SC decoding, only this time assuming that W is a symmetric channel. We first show that the error events $\{\mathscr{E}_i\}$ defined by (4.6) have a symmetry property.

Proposition 17 *For a symmetric B-DMC W, the event $\mathscr{E}_i$ has the property that*

$$(u_1^N, y_1^N) \in \mathscr{E}_i \quad \textit{iff} \quad (a_1^N \oplus u_1^N, a_1^N G_N \cdot y_1^N) \in \mathscr{E}_i \tag{4.13}$$

for each $1 \le i \le N$, $(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N$, $a_1^N \in \mathscr{X}^N$.

Proof. This follows directly from the definition of $\mathscr{E}_i$ by using the symmetry property (4.12) of the channel $W_N^{(i)}$.

Now, consider the transmission of a particular source vector $u_{\mathscr{A}}$ and frozen vector $u_{\mathscr{A}^c}$, jointly forming an input vector u_1^N for the channel W_N. This event is denoted below as $\{U_1^N = u_1^N\}$ instead of the more formal $\{u_1^N\} \times \mathscr{Y}^N$.

Corollary 1 *For a symmetric B-DMC W, for each $1 \le i \le N$ and $u_1^N \in \mathscr{X}^N$, the events $\mathscr{E}_i$ and $\{U_1^N = u_1^N\}$ are independent; hence, $P(\mathscr{E}_i) = P(\mathscr{E}_i \mid \{U_1^N = u_1^N\})$.*

Proof. For $(u_1^N, y_1^N) \in \mathscr{X}^N \times \mathscr{Y}^N$ and $x_1^N = u_1^N G_N$, we have

$$\begin{aligned} P(\mathscr{E}_i \mid \{U_1^N = u_1^N\}) &= \sum_{y_1^N} W_N(y_1^N \mid u_1^N)\, 1_{\mathscr{E}_i}(u_1^N, y_1^N) \\ &= \sum_{y_1^N} W_N(x_1^N \cdot y_1^N \mid 0_1^N)\, 1_{\mathscr{E}_i}(0_1^N, x_1^N \cdot y_1^N) \qquad (4.14) \\ &= P(\mathscr{E}_i \mid \{U_1^N = 0_1^N\}). \qquad (4.15) \end{aligned}$$

Equality follows in (4.14) from (4.11) and (4.13) by taking $a_1^N = u_1^N$, and in (4.15) from the fact that $\{x_1^N \cdot y_1^N : y_1^N \in \mathscr{Y}^N\} = \mathscr{Y}^N$ for any fixed $x_1^N \in \mathscr{X}^N$. The rest of the proof is immediate.

Now, by (4.7), we have, for all $u_1^N \in \mathscr{X}^N$,

$$P(\mathscr{E}_i \mid \{U_1^N = u_1^N\}) \le Z(W_N^{(i)}) \tag{4.16}$$

and, since $\mathscr{E} \subset \cup_{i \in \mathscr{A}} \mathscr{E}_i$, we obtain

$$P(\mathscr{E} \mid \{U_1^N = u_1^N\}) \le \sum_{i \in \mathscr{A}} Z(W_N^{(i)}). \tag{4.17}$$

This implies that, for every symmetric B-DMC W and every $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$ code,

$$P_e(N,K,\mathscr{A},u_{\mathscr{A}^c}) = \sum_{u_{\mathscr{A}}\in\mathscr{X}^K} \frac{1}{2^K} P(\mathscr{E} \mid \{U_1^N = u_1^N\})$$
$$\leq \sum_{i\in\mathscr{A}} Z(W_N^{(i)}). \tag{4.18}$$

This bound on $P_e(N,K,\mathscr{A},u_{\mathscr{A}^c})$ is independent of the frozen vector $u_{\mathscr{A}^c}$. Theorem 4 is now obtained by combining Theorem 2 with Prop. 2, as in the proof of Theorem 3.

Note that although we have given a bound on $P(\mathscr{E}|\{U_1^N = u_1^N\})$ that is independent of u_1^N, we stopped short of claiming that the error event $\mathscr{E}$ is independent of U_1^N because our decision functions $\{h_i\}$ break ties always in favor of $\hat{u}_i = 0$. If this bias were removed by randomization, then $\mathscr{E}$ would become independent of U_1^N.

4.7 Further symmetries of the channel $W_N^{(i)}$

We may use the degrees of freedom in the choice of a_1^N in (4.12) to explore the symmetries inherent in the channel $W_N^{(i)}$. For a given (y_1^N, u_1^i), we may select a_1^N with $a_1^i = u_1^i$ to obtain

$$W_N^{(i)}(y_1^N, u_1^{i-1} \mid u_i) = W_N^{(i)}(a_1^N G_N \cdot y_1^N, 0_1^{i-1} \mid 0). \tag{4.19}$$

So, if we were to prepare a look-up table for the transition probabilities $\{W_N^{(i)}(y_1^N, u_1^{i-1} \mid u_i) : y_1^N \in \mathscr{Y}^N, u_1^i \in \mathscr{X}^i\}$, it would suffice to store only the subset of probabilities $\{W_N^{(i)}(y_1^N, 0_1^{i-1} \mid 0) : y_1^N \in \mathscr{Y}^N\}$.

The size of the look-up table can be reduced further by using the remaining degrees of freedom in the choice of a_{i+1}^N. Let $\mathscr{X}_{i+1}^N \overset{\Delta}{=} \{a_1^N \in \mathscr{X}^N : a_1^i = 0_1^i\}$, $1 \leq i \leq N$. Then, for any $1 \leq i \leq N$, $a_1^N \in \mathscr{X}_{i+1}^N$, and $y_1^N \in \mathscr{Y}^N$, we have

$$W_N^{(i)}(y_1^N, 0^{i-1}|0) = W_N^{(i)}(a_1^N G_N \cdot y_1^N, 0_1^{i-1}|0) \tag{4.20}$$

which follows from (4.19) by taking $u_1^i = 0_1^i$ on the left hand side.

To explore this symmetry further, let $\mathscr{X}_{i+1}^N \cdot y_1^N \overset{\Delta}{=} \{a_1^N G_N \cdot y_1^N : a_1^N \in \mathscr{X}_{i+1}^N\}$. The set $\mathscr{X}_{i+1}^N \cdot y_1^N$ is the *orbit* of y_1^N under the *action group* $\mathscr{X}_{i+1}^N$. The orbits $\mathscr{X}_{i+1}^N \cdot y_1^N$ over variation of y_1^N partition the space $\mathscr{Y}^N$ into equivalence classes. Let $\mathscr{Y}_{i+1}^N$ be a set formed by taking one representative from each equivalence class. The output alphabet of the channel $W_N^{(i)}$ can be represented effectively by the set $\mathscr{Y}_{i+1}^N$.

For example, suppose W is a BSC with $\mathscr{Y} = \{0,1\}$. Each orbit $\mathscr{X}_{i+1}^N \cdot y_1^N$ has 2^{N-i} elements and there are 2^i orbits. In particular, the channel $W_N^{(1)}$ has effectively two outputs, and being symmetric, it has to be a BSC. This is a great simplification since $W_N^{(1)}$ has an apparent output alphabet size of 2^N. Likewise, while $W_N^{(i)}$ has an apparent output alphabet size of 2^{N+i-1}, due to symmetry, the size shrinks to 2^i.

Further output alphabet size reductions may be possible by exploiting other properties specific to certain B-DMCs. For example, if W is a BEC, the channels $\{W_N^{(i)}\}$ are known to be BECs, each with an effective output alphabet size of three.

The symmetry properties of $\{W_N^{(i)}\}$ help simplify the computation of the channel parameters.

Proposition 18 *For any symmetric B-DMC W, the parameters $\{Z(W_N^{(i)})\}$ given by* (1.5) *can be calculated by the simplified formula*

$$Z(W_N^{(i)}) = 2^{i-1} \sum_{y_1^N \in \mathscr{Y}_{i+1}^N} |\mathscr{X}_{i+1}^N \cdot y_1^N| \sqrt{W_N^{(i)}(y_1^N, 0_1^{i-1}|0) W_N^{(i)}(y_1^N, 0_1^{i-1}|1)}.$$

We omit the proof of this result.

For the important example of a BSC, this formula becomes

$$Z(W_N^{(i)}) = 2^{N-1} \sum_{y_1^N \in \mathscr{Y}_{i+1}^N} \sqrt{W_N^{(i)}(y_1^N, 0_1^{i-1}|0)\, W_N^{(i)}(y_1^N, 0_1^{i-1}|1)}.$$

This sum for $Z(W_N^{(i)})$ has 2^i terms, as compared to 2^{N+i-1} terms in (1.5).

Chapter 5
Encoding, Decoding and Construction of Polar Codes

Abstract This chapter considers the encoding, decoding, and construction problems for polar coding.

5.1 Encoding

In this section, we will consider the encoding of polar codes and prove the part of Theorem 5 about encoding complexity. We begin by giving explicit algebraic expressions for G_N, the generator matrix for polar coding, which so far has been defined only in a schematic form by Fig. 3. The algebraic forms of G_N naturally point at efficient implementations of the encoding operation $x_1^N = u_1^N G_N$. In analyzing the encoding operation G_N, we exploit its relation to fast transform methods in signal processing; in particular, we use the bit-indexing idea of [4] to interpret the various permutation operations that are part of G_N.

5.1.1 *Formulas for* G_N

In the following, assume $N = 2^n$ for some $n \geq 0$. Let I_k denote the k-dimensional identity matrix for any $k \geq 1$. We begin by translating the recursive definition of G_N as given by Fig. 3 into an algebraic form:

$$G_N = (I_{N/2} \otimes F)\, R_N\, (I_2 \otimes G_{N/2}), \quad \text{for } N \geq 2,$$

with $G_1 = I_1$.

Either by verifying algebraically that $(I_{N/2} \otimes F)R_N = R_N(F \otimes I_{N/2})$ or by observing that channel combining operation in Fig. 3 can be redrawn equivalently as in Fig. 8, we obtain a second recursive formula

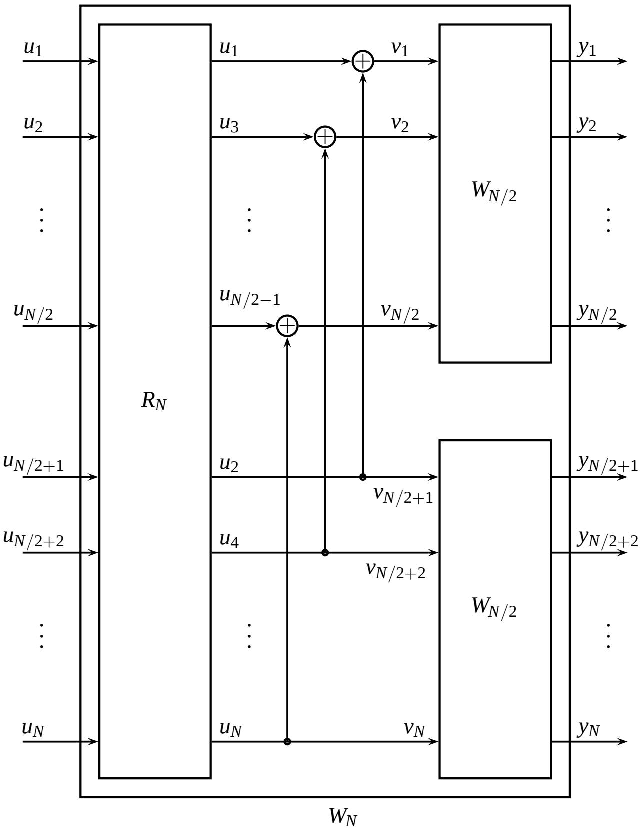

Fig. 5.1 An alternative realization of the recursive construction for W_N.

$$\begin{aligned} G_N &= R_N(F \otimes I_{N/2})(I_2 \otimes G_{N/2}) \\ &= R_N(F \otimes G_{N/2}), \end{aligned} \tag{5.1}$$

valid for $N \geq 2$. This form appears more suitable to derive a recursive relationship. We substitute $G_{N/2} = R_{N/2}(F \otimes G_{N/4})$ back into (5.1) to obtain

$$\begin{aligned} G_N &= R_N\left(F \otimes \left(R_{N/2}\left(F \otimes G_{N/4}\right)\right)\right) \\ &= R_N\left(I_2 \otimes R_{N/2}\right)\left(F^{\otimes 2} \otimes G_{N/4}\right) \end{aligned} \tag{5.2}$$

where (5.2) is obtained by using the identity $(AC) \otimes (BD) = (A \otimes B)(C \otimes D)$ with $A = I_2$, $B = R_{N/2}$, $C = F$, $D = F \otimes G_{N/4}$. Repeating this, we obtain

$$G_N = B_N F^{\otimes n} \tag{5.3}$$

where $B_N \overset{\Delta}{=} R_N(I_2 \otimes R_{N/2})(I_4 \otimes R_{N/4}) \cdots (I_{N/2} \otimes R_2)$. It can seen by simple manipulations that

$$B_N = R_N(I_2 \otimes B_{N/2}). \tag{5.4}$$

We can see that B_N is a permutation matrix by the following induction argument. Assume that $B_{N/2}$ is a permutation matrix for some $N \geq 4$; this is true for $N = 4$ since $B_2 = I_2$. Then, B_N is a permutation matrix because it is the product of two permutation matrices, R_N and $I_2 \otimes B_{N/2}$.

In the following, we will say more about the nature of B_N as a permutation.

5.1.2 Analysis by bit-indexing

To analyze the encoding operation further, it will be convenient to index vectors and matrices with bit sequences. Given a vector a_1^N with length $N = 2^n$ for some $n \geq 0$, we denote its ith element, a_i, $1 \leq i \leq N$, alternatively as $a_{b_1 \cdots b_n}$ where $b_1 \cdots b_n$ is the binary expansion of the integer $i-1$ in the sense that $i = 1 + \sum_{j=1}^n b_j 2^{n-j}$. Likewise, the element A_{ij} of an N-by-N matrix A is denoted alternatively as $A_{b_1 \cdots b_n, b'_1 \cdots b'_n}$ where $b_1 \cdots b_n$ and $b'_1 \cdots b'_n$ are the binary representations of $i-1$ and $j-1$, respectively. Using this convention, it can be readily verified that the product $C = A \otimes B$ of a 2^n-by-2^n matrix A and a 2^m-by-2^m matrix B has elements $C_{b_1 \cdots b_{n+m}, b'_1 \cdots b'_{n+m}} = A_{b_1 \cdots b_n, b'_1 \cdots b'_n} B_{b_{n+1} \cdots b_{n+m}, b'_{n+1} \cdots b'_{n+m}}$.

We now consider the encoding operation under bit-indexing. First, we observe that the elements of F in bit-indexed form are given by $F_{b,b'} = 1 \oplus b' \oplus bb'$ for all $b, b' \in \{0,1\}$. Thus, $F^{\otimes n}$ has elements

$$F^{\otimes n}_{b_1 \cdots b_n, b'_1 \cdots b'_n} = \prod_{i=1}^n F_{b_i, b'_i} = \prod_{i=1}^n (1 \oplus b'_i \oplus b_i b'_i). \tag{5.5}$$

Second, the reverse shuffle operator R_N acts on a row vector u_1^N to replace the element in bit-indexed position $b_1 \cdots b_n$ with the element in position $b_2 \cdots b_n b_1$; that is, if $v_1^N = u_1^N R_N$, then $v_{b_1 \cdots b_n} = u_{b_2 \cdots b_n b_1}$ for all $b_1, \ldots, b_n \in \{0,1\}$. In other words, R_N cyclically rotates the bit-indexes of the elements of a left operand u_1^N to the right by one place.

Third, the matrix B_N in (5.3) can be interpreted as the *bit-reversal* operator: if $v_1^N = u_1^N B_N$, then $v_{b_1 \cdots b_n} = u_{b_n \cdots b_1}$ for all $b_1, \ldots, b_n \in \{0,1\}$. This statement can be proved by induction using the recursive formula (5.4). We give the idea of such a proof by an example. Let us assume that B_4 is a bit-reversal operator and show that the same is true for B_8. Let u_1^8 be any vector over $GF(2)$. Using bit-indexing, it can be written as $(u_{000}, u_{001}, u_{010}, u_{011}, u_{100}, u_{101}, u_{110}, u_{111})$. Since $u_1^8 B_8 = u_1^8 R_8 (I_2 \otimes B_4)$, let us first consider the action of R_8 on u_1^8. The reverse shuffle R_8 rearranges the elements of u_1^8 with respect to odd-even parity of their indices, so $u_1^8 R_8$ equals $(u_{000}, u_{010}, u_{100}, u_{110}, u_{001}, u_{011}, u_{101}, u_{111})$. This has two halves, $c_1^4 \overset{\Delta}{=} (u_{000}, u_{010}, u_{100}, u_{110})$ and $d_1^4 \overset{\Delta}{=} (u_{001}, u_{011}, u_{101}, u_{111})$, corresponding to odd-even index classes. Notice that $c_{b_1 b_2} = u_{b_1 b_2 0}$ and $d_{b_1 b_2} = u_{b_1 b_2 1}$ for all $b_1, b_2 \in \{0,1\}$. This is to be expected since the reverse shuffle rearranges the indices in increasing order within each odd-even index class. Next, consider the action of $I_2 \otimes B_4$ on (c_1^4, d_1^4). The result is $(c_1^4 B_4, d_1^4 B_4)$. By assumption, B_4 is a bit-reversal operation, so $c_1^4 B_4 = (c_{00}, c_{10}, c_{01}, c_{11})$, which in turn equals $(u_{000}, u_{100}, u_{010}, u_{110})$. Likewise, the result of $d_1^4 B_4$ equals $(u_{001}, u_{101}, u_{011}, u_{111})$. Hence, the overall operation B_8 is a bit-reversal operation.

Given the bit-reversal interpretation of B_N, it is clear that B_N is a symmetric matrix, so $B_N^T = B_N$. Since B_N is a permutation, it follows from symmetry that $B_N^{-1} = B_N$.

It is now easy to see that, for any N-by-N matrix A, the product $C = B_N^T A B_N$ has elements $C_{b_1 \cdots b_n, b'_1 \cdots b'_n} = A_{b_n \cdots b_1, b'_n \cdots b'_1}$. It follows that if A is invariant under bit-reversal, i.e., if $A_{b_1 \cdots b_n, b'_1 \cdots b'_n} = A_{b_n \cdots b_1, b'_n \cdots b'_1}$ for every $b_1, \ldots, b_n, b'_1, \ldots, b'_n \in \{0,1\}$, then $A = B_N^T A B_N$. Since $B_N^T = B_N^{-1}$, this is equivalent to $B_N A = A B_T$. Thus, bit-reversal-invariant matrices commute with the bit-reversal operator.

Proposition 19 *For any $N = 2^n$, $n \geq 1$, the generator matrix G_N is given by $G_N = B_N F^{\otimes n}$ and $G_N = F^{\otimes n} B_N$ where B_N is the bit-reversal permutation. G_N is a bit-reversal invariant matrix with*

$$(G_N)_{b_1 \cdots b_n, b'_1 \cdots b'_n} = \prod_{i=1}^{n} (1 \oplus b'_i \oplus b_{n-i} b'_i). \tag{5.6}$$

Proof. $F^{\otimes n}$ commutes with B_N because it is invariant under bit-reversal, which is immediate from (5.5). The statement $G_N = B_N F^{\otimes n}$ was established before; by proving that $F^{\otimes n}$ commutes with B_N, we have established the other statement: $G_N = F^{\otimes n} B_N$. The bit-indexed form (5.6) follows by applying bit-reversal to (5.5).

A fact useful for estimation of minimum Hamming distances of polar codes is the following.

Proposition 20 *For any $N = 2^n$, $n \geq 0$, $b_1, \ldots, b_n \in \{0,1\}$, the rows of G_N and $F^{\otimes n}$ with index $b_1 \cdots b_n$ have the same Hamming weight given by $2^{w_H(b_1, \ldots, b_n)}$.*

Proof. For fixed $b_1, \ldots, b_n$, the sum of the terms $(G_N)_{b_1 \cdots b_n, b'_1 \cdots b'_n}$ (as integers) over all $b'_1, \ldots, b'_n \in \{0,1\}$ gives the Hamming weight of the row of G_N with index $b_1 \cdots b_n$. This sum is easily seen to be $2^{w_H(b_1, \ldots, b_n)}$ where

$$w_H(b_1, \ldots, b_n) \overset{\Delta}{=} \sum_{i=1}^{n} b_i \tag{5.7}$$

is the Hamming weight of $(b_1, \ldots, b_n)$. The proof for $F^{\otimes n}$ is obtained by using the same argument on (5.5).

5.1.3 Encoding complexity

For complexity estimation, our computational model will be a single processor machine with a random access memory. The complexities expressed will be time complexities. The discussion will be given for an arbitrary G_N-coset code with parameters $(N,K,\mathscr{A},u_{\mathscr{A}^c})$.

Let $\chi_E(N)$ denote the worst-case encoding complexity over all $(N,K,\mathscr{A},u_{\mathscr{A}^c})$ codes with a given block-length N. If we take the complexity of a scalar mod-2 addition as 1 unit and the complexity of the reverse shuffle operation R_N as N units, we see from Fig. 3 that $\chi_E(N) \leq N/2+N+2\chi_E(N/2)$. Starting with an initial value $\chi_E(2) = 3$ (a generous figure), we obtain by induction that $\chi_E(N) \leq \frac{3}{2}N\log N$ for all $N = 2^n$, $n \geq 1$. Thus, the encoding complexity is $O(N\log N)$.

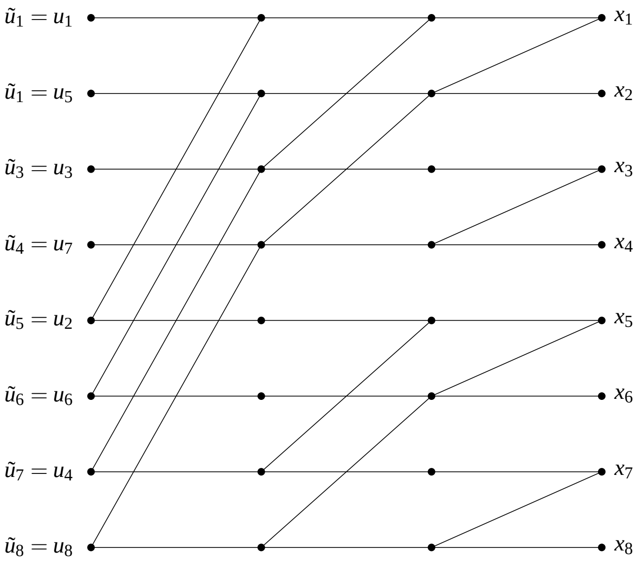

Fig. 5.2 A circuit for implementing the transformation $F^{\otimes 3}$. Signals flow from left to right. Each edge carries a signal 0 or 1. Each node adds (mod-2) the signals on all incoming edges from the left and sends the result out on all edges to the right. (Edges carrying the signals u_i and x_i are not shown.)

A specific implementation of the encoder using the form $G_N = B_N F^{\otimes n}$ is shown in Fig. 9 for $N = 8$. The input to the circuit is the bit-reversed version of u_1^8, i.e., $\tilde{u}_1^8 = u_1^8 B_8$. The output is given by $x_1^8 = \tilde{u}_1^8 F^{\otimes 3} = u_1^8 G_8$. In general, the complexity of this implementation is $O(N\log N)$ with $O(N)$ for B_N and $O(N\log N)$ for $F^{\otimes n}$.

An alternative implementation of the encoder would be to apply u_1^8 in natural index order at the input of the circuit in Fig. 9. Then, we would obtain $\tilde{x}_1^8 = u_1^8 F^{\otimes 3}$

at the output. Encoding could be completed by a post bit-reversal operation: $x_1^8 = \tilde{x}_1^8 B_8 = u_1^8 G_8$.

The encoding circuit of Fig. 9 suggests many parallel implementation alternatives for $F^{\otimes n}$: for example, with N processors, one may do a "column by column" implementation, and reduce the total latency to $\log N$. Various other trade-offs are possible between latency and hardware complexity.

In an actual implementation of polar codes, it may be preferable to use $F^{\otimes n}$ in place of $B_N F^{\otimes n}$ as the encoder mapping in order to simplify the implementation. In that case, the SC decoder should compensate for this by decoding the elements of the source vector u_1^N in bit-reversed index order. We have included B_N as part of the encoder in this paper in order to have a SC decoder that decodes u_1^N in the natural index order, which simplified the notation.

5.2 Decoding

In this section, we consider the computational complexity of the SC decoding algorithm. As in the previous section, our computational model will be a single processor machine with a random access memory and the complexities expressed will be time complexities. Let $\chi_D(N)$ denote the worst-case complexity of SC decoding over all G_N-coset codes with a given block-length N. We will show that $\chi_D(N) = O(N \log N)$.

5.2.1 A first decoding algorithm

Consider SC decoding for an arbitrary G_N-coset code with parameter $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$. Recall that the source vector u_1^N consists of a random part $u_{\mathscr{A}}$ and a frozen part $u_{\mathscr{A}^c}$. This vector is transmitted across W_N and a channel output y_1^N is obtained with probability $W_N(y_1^N|u_1^N)$. The SC decoder observes $(y_1^N, u_{\mathscr{A}^c})$ and generates an estimate $\hat{u}_1^N$ of u_1^N. We may visualize the decoder as consisting of N decision elements (DEs), one for each source element u_i; the DEs are activated in the order 1 to N. If $i \in \mathscr{A}^c$, the element u_i is known; so, the ith DE, when its turn comes, simply sets $\hat{u}_i = u_i$ and sends this result to all succeeding DEs. If $i \in \mathscr{A}$, the ith DE waits until it has received the previous decisions $\hat{u}_1^{i-1}$, and upon receiving them, computes the likelihood ratio (LR)

$$L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \stackrel{\Delta}{=} \frac{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|0)}{W_N^{(i)}(y_1^N, \hat{u}_1^{i-1}|1)}$$

and generates its decision as

$$\hat{u}_i = \begin{cases} 0, & \text{if } L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) \geq 1 \\ 1, & \text{otherwise} \end{cases}$$

which is then sent to all succeeding DEs. This is a single-pass algorithm, with no revision of estimates. The complexity of this algorithm is determined essentially by the complexity of computing the LRs.

A straightforward calculation using the recursive formulas (2.6) and (2.7) gives

$$L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}) = \frac{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})\, L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) + 1}{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) + L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})} \tag{5.8}$$

and

$$L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}) = \left[L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2})\right]^{1-2\hat{u}_{2i-1}} \cdot L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}). \tag{5.9}$$

Thus, the calculation of an LR at length N is reduced to the calculation of two LRs at length $N/2$. This recursion can be continued down to block-length 1, at which point the LRs have the form $L_1^{(1)}(y_i) = W(y_i|0)/W(y_i|1)$ and can be computed directly.

To estimate the complexity of LR calculations, let $\chi_L(k)$, $k \in \{N, N/2, N/4, \ldots, 1\}$, denote the worst-case complexity of computing $L_k^{(i)}(y_1^k, v_1^{i-1})$ over $i \in [1,k]$ and $(y_1^k, v_1^{i-1}) \in \mathscr{Y}^k \times \mathscr{X}^{i-1}$. From the recursive LR formulas, we have the complexity bound

$$\chi_L(k) \leq 2\chi_L(k/2) + \alpha \tag{5.10}$$

where α is the worst-case complexity of assembling two LRs at length $k/2$ into an LR at length k. Taking $\chi_L^{(1)}(y_i)$ as 1 unit, we obtain the bound

$$\chi_L(N) \leq (1+\alpha)N = O(N). \tag{5.11}$$

The overall decoder complexity can now be bounded as $\chi_D(N) \leq K\chi_L(N) \leq N\chi_L(N) = O(N^2)$. This complexity corresponds to a decoder whose DEs do their LR calculations privately, without sharing any partial results with each other. It turns out, if the DEs pool their scratch-pad results, a more efficient decoder implementation is possible with overall complexity $O(N \log N)$, as we will show next.

5.2.2 Refinement of the decoding algorithm

We now consider a decoder that computes the full set of LRs, $\{L_N^{(i)}(y_1^N, \hat{u}_1^{i-1}) : 1 \leq i \leq N\}$. The previous decoder could skip the calculation of $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$ for $i \in \mathscr{A}^c$; but now we do not allow this. The decisions $\{\hat{u}_i : 1 \leq i \leq N\}$ are made in exactly the same manner as before; in particular, if $i \in \mathscr{A}^c$, the decision $\hat{u}_i$ is set to the known frozen value u_i, regardless of $L_N^{(i)}(y_1^N, \hat{u}_1^{i-1})$.

To see where the computational savings will come from, we inspect (5.8) and (5.9) and note that each LR value in the pair

$$(L_N^{(2i-1)}(y_1^N, \hat{u}_1^{2i-2}), L_N^{(2i)}(y_1^N, \hat{u}_1^{2i-1}))$$

is assembled from the same pair of LRs:

$$(L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}), L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2})).$$

Thus, the calculation of all N LRs at length N requires exactly N LR calculations at length $N/2$.[1] Let us split the N LRs at length $N/2$ into two classes, namely,

$$\begin{aligned} &\{L_{N/2}^{(i)}(y_1^{N/2}, \hat{u}_{1,o}^{2i-2} \oplus \hat{u}_{1,e}^{2i-2}) : 1 \leq i \leq N/2\}, \\ &\{L_{N/2}^{(i)}(y_{N/2+1}^N, \hat{u}_{1,e}^{2i-2}) : 1 \leq i \leq N/2\}. \end{aligned} \tag{5.12}$$

Let us suppose that we carry out the calculations in each class independently, without trying to exploit any further savings that may come from the sharing of LR values between the two classes. Then, we have two problems of the same type as the original but at half the size. Each class in (5.12) generates a set of $N/2$ LR calculation requests at length $N/4$, for a total of N requests. For example, if we let $\hat{v}_1^{N/2} \overset{\Delta}{=} \hat{u}_{1,o}^{N/2} \oplus \hat{u}_{1,e}^{N/2}$, the requests arising from the first class are

$$\begin{aligned} &\{L_{N/4}^{(i)}(y_1^{N/4}, \hat{v}_{1,o}^{2i-2} \oplus \hat{v}_{1,e}^{2i-2}) : 1 \leq i \leq N/4\}, \\ &\{L_{N/4}^{(i)}(y_{N/4+1}^{N/2}, \hat{v}_{1,e}^{2i-2}) : 1 \leq i \leq N/4\}. \end{aligned}$$

Using this reasoning inductively across the set of all lengths $\{N, N/2, \ldots, 1\}$, we conclude that the total number of LRs that need to be calculated is $N(1 + \log N)$.

So far, we have not paid attention to the exact order in which the LR calculations at various block-lengths are carried out. Although this gave us an accurate count of the total number of LR calculations, for a full description of the algorithm, we need to specify an order. There are many possibilities for such an order, but to be specific we will use a depth-first algorithm, which is easily described by a small example.

[1] Actually, some LR calculations at length $N/2$ may be avoided if, by chance, some duplications occur, but we will disregard this.

We consider a decoder for a code with parameter $(N, K, \mathscr{A}, u_{\mathscr{A}^c})$ chosen as $(8, 5, \{3, 5, 6, 7, 8\}, (0, 0, 0)\}$. The computation for the decoder is laid out in a graph as shown in Fig. 10. There are $N(1 + \log N) = 32$ nodes in the graph, each responsible for computing an LR request that arises during the course of the algorithm. Starting from the left-side, the first column of nodes correspond to LR requests at length 8 (decision level), the second column of nodes to requests at length 4, the third at length 2, and the fourth at length 1 (channel level).

Each node in the graph carries two labels. For example, the third node from the bottom in the third column has the labels $(y_5^6, \hat{u}_2 \oplus \hat{u}_4)$ and 26; the first label indicates that the LR value to be calculated at this node is $L_8^{(2)}(y_5^6, \hat{u}_2 \oplus \hat{u}_4)$ while the second label indicates that this node will be the 26th node to be activated. The numeric labels, 1 through 32, will be used as quick identifiers in referring to nodes in the graph.

The decoder is visualized as consisting of N DEs situated at the left-most side of the decoder graph. The node with label $(y_1^8, \hat{u}_1^{i-1})$ is associated with the ith DE, $1 \leq i \leq 8$. The positioning of the DEs in the left-most column follows the bit-reversed index order, as in Fig. 9.

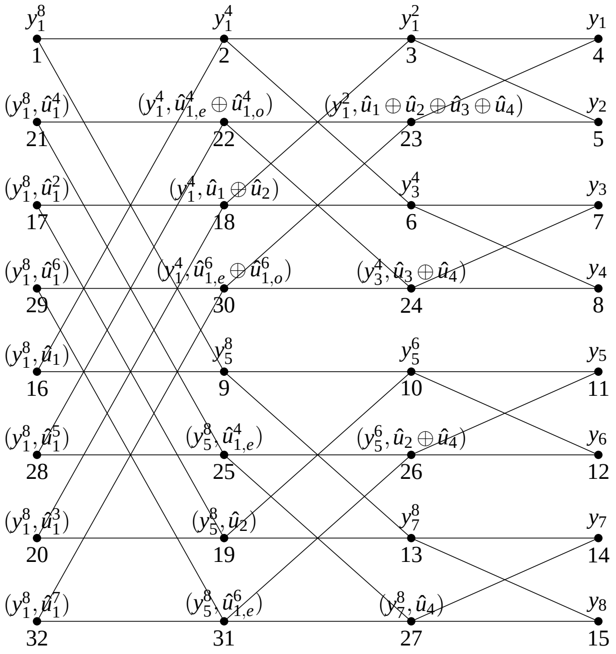

Fig. 5.3 An implementation of the successive cancellation decoder for polar coding at block-length $N = 8$.

Decoding begins with DE 1 activating node 1 for the calculation of $L_8^{(1)}(y_1^8)$. Node 1 in turn activates node 2 for $L_4^{(1)}(y_1^4)$. At this point, program control passes to node 2, and node 1 will wait until node 2 delivers the requested LR. The process continues. Node 2 activates node 3, which activates node 4. Node 4 is a node at the channel level; so it computes $L_1^{(1)}(y_1)$ and passes it to nodes 3 and 23, its left-side neighbors. In general a node will send its computational result to all its left-side neighbors (although this will not be stated explicitly below). Program control will be passed back to the left neighbor from which it was received.

Node 3 still needs data from the right side and activates node 5, which delivers $L_1^{(1)}(y_2)$. Node 3 assembles $L_2^{(1)}(y_1^2)$ from the messages it has received from nodes 4 and 5 and sends it to node 2. Next, node 2 activates node 6, which activates nodes 7 and 8, and returns its result to node 2. Node 2 compiles its response $L_4^{(1)}(y_1^4)$ and sends it to node 1. Node 1 activates node 9 which calculates $L_4^{(1)}(y_5^8)$ in the same manner as node 2 calculated $L_4^{(1)}(y_1^4)$, and returns the result to node 1. Node 1 now assembles $L_8^{(1)}(y_1^8)$ and sends it to DE 1. Since u_1 is a frozen node, DE 1 ignores the received LR, declares $\hat{u}_1 = 0$, and passes control to DE 2, located next to node 16.

DE 2 activates node 16 for $L_8^{(2)}(y_1^8, \hat{u}_1)$. Node 16 assembles $L_8^{(2)}(y_1^8, \hat{u}_1)$ from the already-received LRs $L_4^{(1)}(y_1^4)$ and $L_4^{(1)}(y_5^8)$, and returns its response without activating any node. DE 2 ignores the returned LR since u_2 is frozen, announces $\hat{u}_2 = 0$, and passes control to DE 3.

DE 3 activates node 17 for $L_8^{(3)}(y_1^8, \hat{u}_1^2)$. This triggers LR requests at nodes 18 and 19, but no further. The bit u_3 is not frozen; so, the decision $\hat{u}_3$ is made in accordance with $L_8^{(3)}(y_1^8, \hat{u}_1^2)$, and control is passed to DE 4. DE 4 activates node 20 for $L_8^{(4)}(y_1^8, \hat{u}_1^3)$, which is readily assembled and returned. The algorithm continues in this manner until finally DE 8 receives $L_8^{(7)}(y_1^8, \hat{u}_1^7)$ and decides $\hat{u}_8$.

There are a number of observations that can be made by looking at this example that should provide further insight into the general decoding algorithm. First, notice that the computation of $L_8^{(1)}(y_1^8)$ is carried out in a subtree rooted at node 1, consisting of paths going from left to right, and spanning all nodes at the channel level. This subtree splits into two disjoint subtrees, namely, the subtree rooted at node 2 for the calculation of $L_4^{(1)}(y_1^4)$ and the subtree rooted at node 9 for the calculation of $L_4^{(1)}(y_5^8)$. Since the two subtrees are disjoint, the corresponding calculations can be carried out independently (even in parallel if there are multiple processors). This splitting of computational subtrees into disjoint subtrees holds for all nodes in the graph (except those at the channel level), making it possible to implement the decoder with a high degree of parallelism.

Second, we notice that the decoder graph consists of *butterflies* (2-by-2 complete bipartite graphs) that tie together adjacent levels of the graph. For example, nodes 9, 19, 10, and 13 form a butterfly. The computational subtrees rooted at nodes 9 and 19 split into a single pair of computational subtrees, one rooted at node 10, the other at node 13. Also note that among the four nodes of a butterfly, the upper-left node is always the first node to be activated by the above depth-first algorithm and the lower-left node always the last one. The upper-right and lower-right nodes are activated by the upper-left node and they may be activated in any order or even in parallel. The algorithm we specified always activated the upper-right node first, but this choice was arbitrary. When the lower-left node is activated, it finds the LRs from its right neighbors ready for assembly. The upper-left node assembles the LRs it receives from the right side as in formula (5.8), the lower-left node as in (5.9). These formulas show that the butterfly patterns impose a constraint on the completion time of LR calculations: in any given butterfly, the lower-left node needs to wait for the result of the upper-left node which in turn needs to wait for the results of the right-side nodes.

Variants of the decoder are possible in which the nodal computations are scheduled differently. In the "left-to-right" implementation given above, nodes waited to be activated. However, it is possible to have a "right-to-left" implementation in which each node starts its computation autonomously as soon as its right-side neighbors finish their calculations; this allows exploiting parallelism in computations to the maximum possible extent.

For example, in such a fully-parallel implementation for the case in Fig. 10, all eight nodes at the channel-level start calculating their respective LRs in the first time slot following the availability of the channel output vector y_1^8. In the second time slot, nodes 3, 6, 10, and 13 do their LR calculations in parallel. Note that this is the maximum degree of parallelism possible in the second time slot. Node 23, for example, cannot calculate $L_N^{(2)}(y_1^2, \hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4)$ in this slot, because $\hat{u}_1 \oplus \hat{u}_2 \oplus \hat{u}_3 \oplus \hat{u}_4$ is not yet available; it has to wait until decisions $\hat{u}_1$, $\hat{u}_2$, $\hat{u}_3$, $\hat{u}_4$ are announced by the corresponding DEs. In the third time slot, nodes 2 and 9 do their calculations. In time slot 4, the first decision $\hat{u}_1$ is made at node 1 and broadcast to all nodes across the graph (or at least to those that need it). In slot 5, node 16 calculates $\hat{u}_2$ and broadcasts it. In slot 6, nodes 18 and 19 do their calculations. This process continues until time slot 15 when node 32 decides $\hat{u}_8$. It can be shown that, in general, this fully-parallel decoder implementation has a latency of $2N-1$ time slots for a code of block-length N.

5.3 Code construction

The original polar coding paper [1] left the polar coding construction problem unsolved. Only for the BEC, a solution was given. For the general case, a Monte Carlo simulation method was suggested. Although the problem looked very formidable, rapid progress has been made in this area starting with Mori and Tanaka [10] who proposed a density evolution approach but did not address the numerical problems in computing the densities with sufficient precision. A major advance was made by Tal and Vardy [16] who exploited the notions of channel degradation and "upgradation" to provide not just approximations but also upper and lower bounds on the channel parameters, such as $I(W_N^{(i)})$ and $Z(W_N^{(i)})$, that are involved in code construction. This line of work has been extended in Pedarsani *et al.* [12] where specific bounds on the approximation error were derived. The presentation below follows largely [12] and Şaşoğlu [5].

For polar code construction, we seek an algorithm that accepts as input a triple (W,N,K) where W is the B-DMC on which the code will be used, N is the code block-length, and K is the dimensionality of the code and produces as output an information set $\mathscr{A} \subset \{1,\dots,N\}$ of size K such that $\sum_{i\in\mathscr{A}} Z(W_N^{(i)})$ is as small as possible. Finding a good frozen vector $u_{\mathscr{A}^c}$ should also be included as part of the desired output of a code construction algorithm in general. However, if W is a symmetric channel then the code performance is not affected by the choice of $u_{\mathscr{A}^c}$ and this second issue disappears. The following discussion is restricted to symmetric channels and we will exclude finding a good frozen vector from the code construction problem. We use the abbreviation BMS to refer to binary-input memoryless symmetric channels. The output alphabet for a BMS will be assumed finite but the methods here applicable to BMS channels with a continuous output alphabet such as binary-input additive Gaussian noise channels.

In principle, the code construction problem can be solved by computing the transition probabilities of all the channels $\{W_{2^{n-k}}^{(i)} : 0 \le k \le n, 1 \le i \le 2^{n-k}\}$ created through the course of the polarization construction, as depicted in Fig. 3.1. Such a computation would use the recursive relations given in Proposition 3 starting with $W_1^{(1)} = W$. Altogether there are $2N-1$ channels in this collection and it may appear that this calculation should have complexity $O(N)$ where $N = 2^n$ is the code block length. Unfortunately, this computation is complicated by the exponentially growing size of the output spaces of the channels involved. For example, the output of the channel $W_N^{(i)}$ is the vector $y^N u^{i-1}$ which can take on $M^N 2^{i-1}$ possible values if W is a channel with M outputs.

There is an exceptional case where the above recursive calculation is feasible. If W is a BEC, each channel in the collection $\{W_{2^{n-k}}^{(i)}\}$ is a BEC and the erasure probabilities can be calculated using the recursive formulas (2.23) with overall complexity $O(N)$. Although the channels created from a BEC W also appear to have an exponentially growing size for their output spaces, after merging equivalent output letters, only three letters remain: 0,1, and erasure. The BEC example suggests that merging similar output letters may lead to a low-complexity approximate code construction algorithm for general channels. This is indeed the key idea of the methods that will be presented in the rest of this section.

Before we present the specific methods for polar code construction we need to develop some general results about BMS channels.

5.3.1 A general representation of BMS channels

Definition 1 *A channel $W : \mathscr{X} \to \mathscr{Y}$ is said to be the sum of channels $\{W_i : 1 \le i \le M\}$ with weights $\{p_i : 1 \le i \le M\}$ if the following hold:*

- *$\{p_i : 1 \le i \le M\}$ is a probability distribution*

- *The channels entering into the sum have the form*

$$W_i : \mathscr{X} \to \mathscr{Y}_i$$

with the output alphabets $\mathscr{Y}_i$, $1 \leq i \leq M$, *forming a partition of the output alphabet* $\mathscr{Y}$ *of the original channel:*

$$\mathscr{Y} = \bigcup_{i=1}^{M} \mathscr{Y}_i, \qquad \mathscr{Y}_i \cap \mathscr{Y}_j = \emptyset,\ i \neq j.$$

- *The transition probabilities are related by*

$$W(y|x) = p_i W_i(y|x), \qquad \textit{whenever } y \in \mathscr{Y}_i,\ 1 \leq i \leq M.$$

We write $W = \sum_{i=1}^{M} p_i W_i$ *to denote that* W *is a sum of channels in this sense.*

Proposition 21 *Any BMS channel* $W : \{0,1\} \to \mathscr{Y}$ *with a finite output alphabet can be written as the sum of BSCs:*

$$W = \sum_{i=1}^{M} p_i BSC(\varepsilon_i),$$

where the crossover probabilities ε_i *are between 0 and 1/2.*

Proof. Since W is symmetric, for each output letter y there exists a conjugate letter $\bar{y}$ so that $W(y|0) = W(\bar{y}|1)$ and $W(y|1) = W(\bar{y}|0)$. Thus, each output letter, together with its conjugate $\bar{y}$ defines a BSC with input alphabet $\{0,1\}$ and output alphabet $\{y, \bar{y}\}$. Some of these BSCs may have identical crossover probabilities; in that case, we merge the BSCs with identical crossover probabilities into a single BSC. Output symbols y for which $W(y|0) = W(y|1)$ (which are effectively erasures) may be split into two symbols if necessary to represent them as a BSC with crossover probability 1/2.

Example 1 *A binary erasure channel* W *with erasure probability* ε *can be written as* $W = (1-\varepsilon)BSC(0) + \varepsilon BSC(1/2)$.

It will be convenient to generalize the above definitions to the case where the channel output alphabet can be continuous. In this more general case, we may represent any BMS channel W in the form

$$W = \int_0^{1/2} f(\varepsilon)\mathrm{BSC}(\varepsilon)\, d\varepsilon$$

where f is a pdf on $[0, 1/2]$. This representation covers the previous one by taking $f(\varepsilon) = \sum_{i=1}^{M} p_i \delta(\varepsilon - \varepsilon_i)$.

Given the characterization of a BMS channel W as a sum of BSCs, it is easy to see that the symmetric capacity $I(W)$ and the Bhattacharyya parameter $Z(W)$ can be calculated as

$$I(W) = \int_0^{1/2} f(\varepsilon)[1 - \mathscr{H}(\varepsilon)]\, d\varepsilon$$

and

$$Z(W) = \int_0^{1/2} f(\varepsilon)\sqrt{4\varepsilon(1-\varepsilon)}\, d\varepsilon.$$

These parameters may alternatively be denoted as $I(f)$ and $Z(f)$.

5.3.2 Channel approximation

A given BMS channel W may be approximated for a given purpose by suitably approximating its characterizing pdf f. In polar coding, typically, we wish to replace a given f with a simpler f' while keeping the approximation error, as measured by $|I(f) - I(f')|$ or $|Z(f) - Z(f')|$, small. Since both $I(f)$ and $Z(f)$ are continuous functions of f taking values in a closed compact interval (namely, $[0,1]$), this approximation problem can be solved without much difficulty. For our purposes it will be sufficient to use the following simple “quantizer” for approximating BMS channels.

Proposition 22 *Let $L \geq 1$ be a fixed integer. For $i = 0, 1, \ldots, L$, let $\delta_i \in [0, 1/2]$ be (the unique real number) such that a BSC with crossover probability δ_i has symmetric capacity $1 - (i/L)$, i.e., $\mathscr{H}(\delta_i) = i/L$. Let W be a symmetric binary-input memoryless channel characterized by a PDF f. Let $\tilde{W}$ be the channel*

$$\tilde{W} = \sum_{i=0}^{L} \tilde{p}_i BSC(\delta_i)$$

where

$$\tilde{p}_i = \int_{\delta_{i-1}}^{\delta_i} f(\delta) d\delta, \qquad i = 1, \ldots, L.$$

(The integrals are over $[\delta_{i-1}, \delta_i)$ except for the last one which is over $[\delta_{L-1}, \delta_L]$.) Then, $I(\tilde{W}) \leq I(W) \leq I(\tilde{W}) + 1/L$.

Proof. Since $\mathscr{H}(\delta)$ is an increasing function of δ in the interval $[0, 1/2]$, we have $0 = \delta_0 < \delta_1 < \cdots < \delta_L = 1/2$. Thus, these points partition $[0, 1/2]$ into disjoint quantization intervals. The first half of the desired inequality is obtained as

$$\begin{aligned} I(W) &= \int_0^{1/2} f(\delta)[1 - \mathscr{H}(\delta)] d\delta \\ &= \sum_{i=1}^{L} \int_{\delta_{i-1}}^{\delta_i} f(\delta)[1 - \mathscr{H}(\delta)] d\delta \\ &\geq \sum_{i=1}^{L} \int_{\delta_{i-1}}^{\delta_i} f(\delta)[1 - \mathscr{H}(\delta_i)] d\delta \end{aligned}$$

$$= I(\tilde{W})$$

where the inequality uses the monotone increasing property of $\mathscr{H}(\delta)$ for $\delta \in [0,1/2]$. To obtain the second half, we use the monotone property again but in the reverse direction.

$$\begin{aligned} I(W) &\le \sum_{i=1}^{L} \int_{\delta_{i-1}}^{\delta_i} f(\delta)[1 - \mathscr{H}(\delta_{i-1})]d\delta \\ &= \sum_{i=1}^{L} p_i[1-(i-1)/L] \\ &= I(\tilde{W}) + 1/L. \end{aligned}$$

We will show that the above type of quantization creates a degraded channel in the following sense.

Definition 2 *Let $W : \mathscr{X} \to \mathscr{Y}$ and $W' : \mathscr{X} \to \mathscr{Y}'$ be two channels. We say that W' is degraded wrt W if there exists a third channel $P : \mathscr{Y} \to \mathscr{Y}'$ such that*

$$W'(y'|x) = \sum_y P(y'|y)W(y|x).$$

We write $W' \preceq W$ to indicate that W' is degraded wrt W.

Proposition 23 *Let W be a BMS channel and $\tilde{W}$ be its quantized version as above. Then, $\tilde{W} \preceq W$.*

Proof. We may represent the quantizer as a channel (a deterministic one).

Proposition 24 *Let W and W' be two B-DMCs with $W \preceq W'$. Then, $I(W) \le I(W')$ and $Z(W) \ge Z(W')$. Furthermore, channel degradedness relationship propagates through the polarization construction in the sense that*

$$W_N^{(i)} \preceq (W')_N^{(i)}, \qquad \text{for all } N = 2^n,\, 1 \le i \le N.$$

Corollary 2 *Let $W_2^{(1)}$ and $W_2^{(2)}$ be the channels obtained from W by one-step polarization. Similarly let $\tilde{W}_2^{(1)}$ and $\tilde{W}_2^{(2)}$ be obtained from the quantized channel $\tilde{W}$. Then,*

$$I(\tilde{W}_2^{(1)}) \le I(W_2^{(1)}) \quad \text{and} \quad I(\tilde{W}_2^{(2)}) \le I(W_2^{(2)}).$$

5.3.3 A code construction algorithm

We have completed intoducing the basic notions that underly the code construction algorithm that follows. Let W be a given BMS and let $\tilde{W}$ be a downward quantization of W with resolution L as defined above. From the identities

$$I(W_2^{(1)}) + I(W_2^{(2)}) = 2I(W)$$

and

$$I(\tilde{W}_2^{(1)}) + I(\tilde{W}_2^{(2)}) = 2I(\tilde{W})$$

we obtain

$$[I(W_2^{(1)}) - I(\tilde{W}_2^{(1)})] + [I(W_2^{(2)}) - I(\tilde{W}_2^{(2)})] = 2[I(W) - I(\tilde{W})]$$

This shows that the average approximation error after one-step polarization is the same as the error before the polarization step. Since the two difference terms on the left are non-negative (channel degradedness) and the difference term on the right is bounded by $1/L$, we have

$$|I(W_2^{(1)}) - I(\tilde{W}_2^{(1)})| + |I(W_2^{(2)}) - I(\tilde{W}_2^{(2)}) \leq 2/L.$$

Thus, the average *absolute* error is also bounded by $2/L$. The fact that we have a bound on the absolute error is essential for the final result.,

While the quantized channel $\tilde{W}$ has at most $2(L+1)$ output letters, the channels $\tilde{W}_2^{(1)}$ and $\tilde{W}_2^{(2)}$ have many more output letters. The idea of low-complexity polar code construction is to quantize the channels $\tilde{W}_2^{(i)}$ again before continuing with the next step of polarization. The method can be described more precisely by referring to Fig. 3.1 again. The quantization procedure replaces the root node by $\tilde{W}$ before applying the first polarization step. The two channels created at level 1 are now $\tilde{W}_2^{(1)}$ and $\tilde{W}_2^{(2)}$. Before continuing further, these channels are quantized to resolution L and polarization is applied to obtain the four channels at level 2. We shall abuse the notation to denote by $\{\tilde{W}_{2^{n-k}}^{(i)} : 0 \leq k \leq n, 1 \leq i \leq 2^{n-k}\}$ the channels obtained in the course of this quantize-polarize procedure. Each branching point in Fig. 3.1 causes an incremental quantization error. The average quantization error at each node is bounded by $1/L$. An inductive argument shows that the overall average absolute quantization error at level k of this procedure is bounded as

$$\frac{1}{2^{n-k}} \sum_{i=1}^{2^{n-k}} |I(W_{2^{n-k}}^{(i)} - I(\tilde{W}_{2^{n-k}}^{(i)})| \leq k/L, \qquad k = 1, \ldots, n. \tag{5.13}$$

In particular, the average absolute quantization error at the last level is bounded by n/L. We conclude by Markov's inequality that at least a fraction $1 - \sqrt{n/L}$ of the quantities $\{I(W_N^{(i)}) : 1 \leq i \leq N\}$ are computed with an error not exceeding $\sqrt{n/L}$. (It is here that having a bound on average absolute error is crucial.) By taking $L = n^2$, one can ensure that, with the exception of at most a fraction $1/\sqrt{n}$, the terms $\{I(W_N^{(i)})\}$ are computed with an error not exceeding $1/\sqrt{n}$. This means that with a negligible loss in rate we can identify the good coordinates. The overall complexity of this calculation is roughly $O(L^2 N)$ or $O(Nn^2)$ if L is chosen as n^2.

References

1. Arıkan, E.: Channel polarization: A method for constructing capacity-achieving codes for symmetric binary-input memoryless channels. IEEE Trans. Inform. Theory **IT-55**(7), 3051–3073 (2009)
2. Arıkan, E., Telatar, E.: On the rate of channel polarization. In: Proc. 2009 IEEE Int. Symp. Inform. Theory, pp. 1493–1495. Seoul, S. Korea (2009)
3. Chung, K.L.: A Course in Probability Theory, 2nd ed. Academic: New York (1974)
4. Cooley, J.W., Tukey, J.W.: An algorithm for the machine calculation of complex Fourier series. *Math. Comput.* **19**(90), 297–301 (1965)
5. Şaşoğlu, E.: Polarization and polar coding (Spring 2012). Unpublished notes (to appear in the "Foundation Series."
6. Gallager, R.G.: Information Theory and Reliable Communication. Wiley: New York (1968)
7. Hassani, S.H., Mori, R., Tanaka, T., Urbanke, R.: Rate-dependent analysis of the asymptotic behavior of channel polarization (Oct. 2011). ArXiv.org:1110.0194
8. Hassani, S.H., Urbanke, R.: On the scaling of polar codes: I. the behavior of polarized channels. In: Proc. 2010 IEEE Int. Symp. Inform. Theory, pp. 874–878. Austin, TX (2010)
9. Lin, S., Costello, Jr., D.J.: Error Control Coding, (2nd ed). Pearson: N.J. (2004)
10. Mori, R., Tanaka, T.: Performance of polar codes with the construction using density evolution. IEEE Communications Letters **13**(7), 519–521 (2009)
11. Muller, D.E.: Application of boolean algebra to switching circuit design and to error correction. IRE Trans. Electronic Computers **EC-3**, 6–12 (1954)
12. Pedarsani, R., Hassani, S.H., Tal, I., Telatar, E.: On the construction of polar codes. In: Proc. 2011 IEEE Int. Symp. Inform. Theory, pp. 11–15. St. Petersburg, Russia (2011)
13. Plotkin, M.: Binary codes with specified minimum distance. IRE Trans. Inform. Theory **6**(4), 445–450 (1960)
14. Reed, I.: A class of multiple-error-correcting codes and the decoding scheme. IRE Trans. Inform. Theory **4**(4), 39–44 (1954)
15. Shannon, C.E.: A mathematical theory of communication. Bell System Tech. J. **27**(2), 379–423, 623–656 (1948)
16. Tal, I., Vardy, A.: How to construct polar codes (May 2011). ArXiv.org:1105.6164
17. Tanaka, T.: On speed of channel polarization. In: Proc. 2010 IEEE Information Theory Workshop, pp. 1–5. Dublin, Ireland (2010)
18. Tanaka, T., Mori, R.: Refined rate of channel polarization. In: Proc. 2010 IEEE Int. Symp. Inform. Theory, pp. 889–893. Austin, TX (2010)