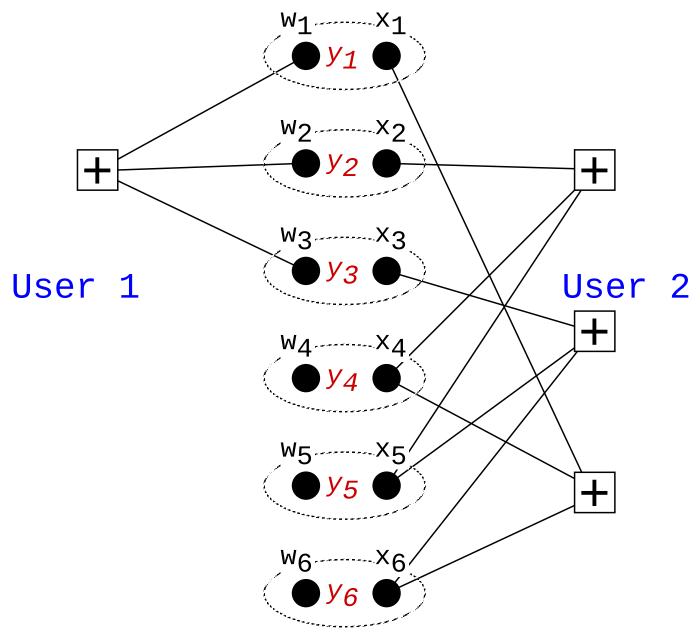
Turbolike Codes on Nonstandard Channel Models

Robert J. McEliece

California Institute of Technology

Pasadena, California, USA

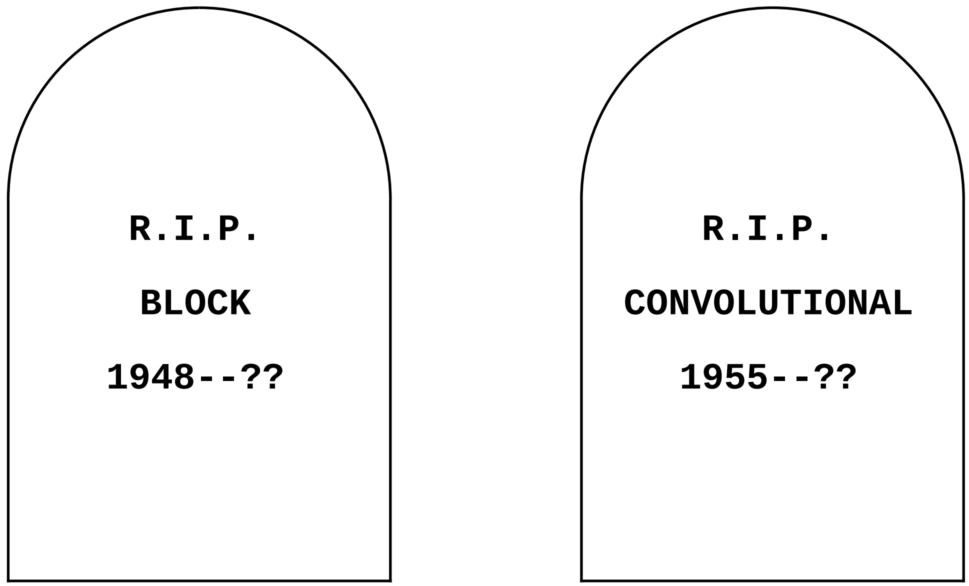


2001 International Symposium on Information Theory

Washington D.C., June 25, 2001

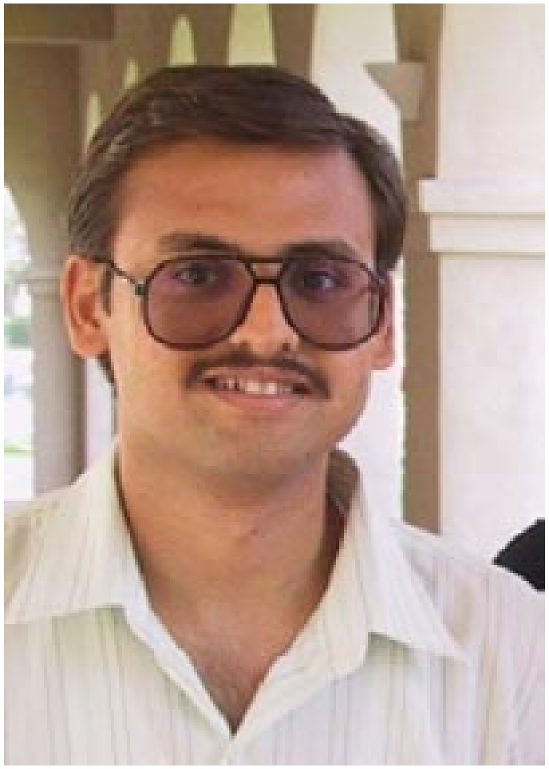
Is Coding Dead?

Robert J. McEliece
California Institute of Technology
Pasadena, California, USA

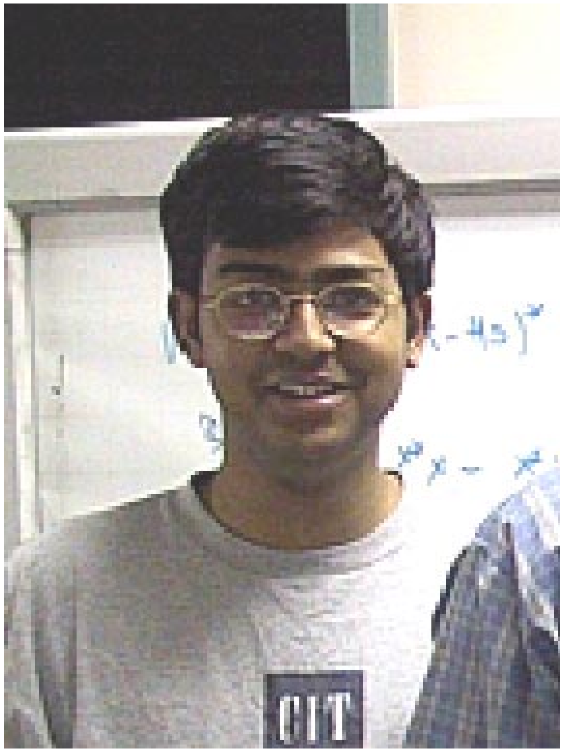


2001 International Symposium on Information Theory
Washington D.C., June 25, 2001

Special Thanks To



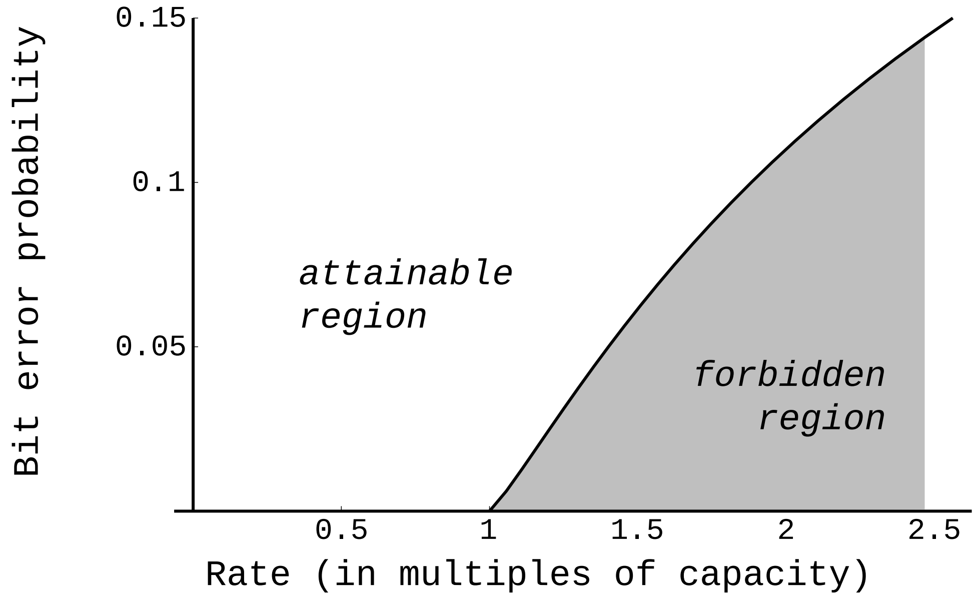
Aamod Khandekar



Ravi Palanki

(Who did most of the work)

Shannon's Channel Coding Theorem



Theorem: *For any (discrete-input memoryless) channel, there exists a number C, the channel capacity, such that for any desired data rate $R < C$ and any desired error probability $p > 0$, it is possible to design an encoder-decoder pair that permits the transmission of data over the channel at rate R and decoded error probability $< p$.*

How Hard is it to Approach Channel Capacity?

How Hard is it to Approach Channel Capacity?

- Desired transmission rate $R = C(1 - \epsilon)$.

How Hard is it to Approach Channel Capacity?

- Desired transmission rate $R = C(1 - \epsilon)$.
- Desired decoder error probability $= p$.

How Hard is it to Approach Channel Capacity?

- Desired transmission rate $R = C(1-\epsilon)$.
- Desired decoder error probability $= p$.
- $\chi_E(\epsilon, p) =$ the minimum possible *encoding* complexity, in operations *per information bit*.

How Hard is it to Approach Channel Capacity?

- Desired transmission rate $R = C(1 - \epsilon)$.
- Desired decoder error probability $= p$.
- $\chi_E(\epsilon, p) =$ the minimum possible *encoding* complexity, in operations *per information bit*.
- $\chi_D(\epsilon, p) =$ the minimum possible *decoding* complexity, in operations *per information bit*.

How Hard is it to Approach Channel Capacity?

- Desired transmission rate $R = C(1 - \epsilon)$.
- Desired decoder error probability $= p$.
- $\chi_E(\epsilon, p) =$ the minimum possible *encoding* complexity, in operations *per information bit*.
- $\chi_D(\epsilon, p) =$ the minimum possible *decoding* complexity, in operations *per information bit*.

For fixed p, how do $\chi_E(\epsilon, p)$ and $\chi_D(\epsilon, p)$, behave, as $\epsilon \to 0$?

The Classical Results.

Theorem: *On a discrete memoryless channel of capacity* C, *for any fixed* $p > 0$, *as* $\epsilon \to 0$,

$$\chi_E(\epsilon, p) = O(1/\epsilon^2)$$

$$\chi_D(\epsilon, p) = 2^{O(1/\epsilon^2)}.$$

The Classical Results.

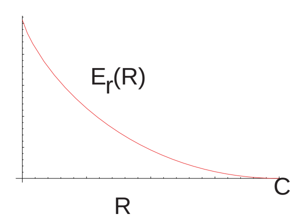
Theorem: *On a discrete memoryless channel of capacity C, for any fixed $p > 0$, as $\epsilon \longrightarrow 0$,*

$$\chi_E(\epsilon, p) = O(1/\epsilon^2)$$

$$\chi_D(\epsilon, p) = 2^{O(1/\epsilon^2)}.$$

Proof: Use linear codes with (per-bit) encoding complexity $O(n)$, and ML decoding with complexity $2^{O(n)}$. And $n = O(1/\epsilon^2)$, because of the random coding exponent:

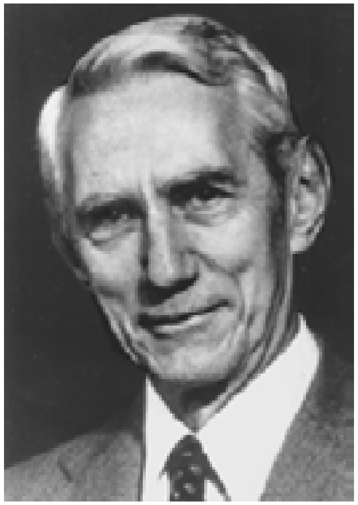
$$\overline{p} \leq e^{-nE_r(R)}$$



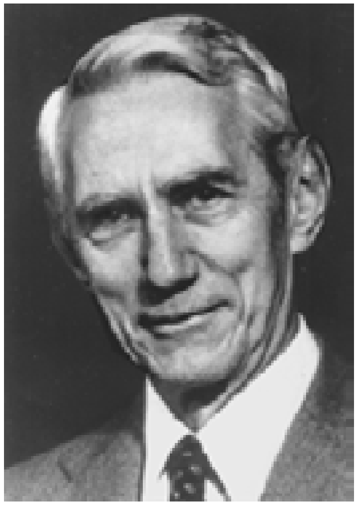
where

$$E_r(C(1-\epsilon)) \approx K\epsilon^2 \qquad \text{as } \epsilon \longrightarrow 0. \qquad \blacksquare$$

The Shannon Challenge



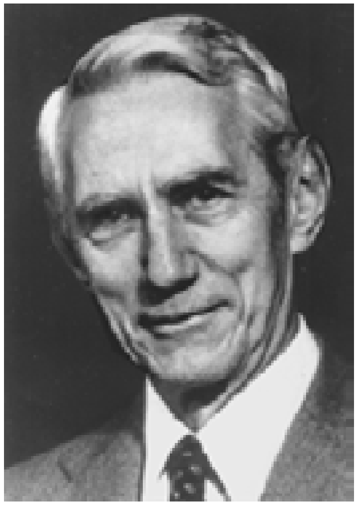
The Shannon Challenge



- *For mathematicians: Reduce the decoding complexity to*

$$\chi_D(\epsilon, p) = O((\frac{1}{\epsilon})^m).$$

The Shannon Challenge



- *For mathematicians: Reduce the decoding complexity to*

$$\chi_D(\epsilon, p) = O((\frac{1}{\epsilon})^m).$$

- *For engineers: Approach the Shannon limit practically!*

Has Shannon's Challenge Already Been Met by Turbo-Codes?

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)
- *Asymmetric Turbo* Codes (Costello and Massey)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)
- *Asymmetric Turbo* Codes (Costello and Massey)
- *Mixture Turbo* Codes (Divsalar, Dolinar, and Pollara)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)
- *Asymmetric Turbo* Codes (Costello and Massey)
- *Mixture Turbo* Codes (Divsalar, Dolinar, and Pollara)
- *Doped Turbo* Codes (ten Brink)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

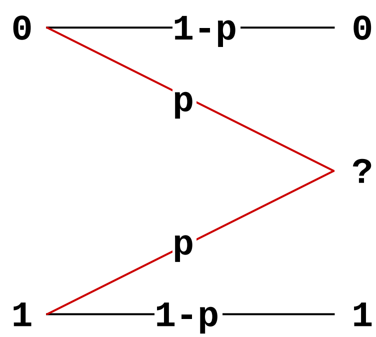
- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)
- *Asymmetric Turbo* Codes (Costello and Massey)
- *Mixture Turbo* Codes (Divsalar, Dolinar, and Pollara)
- *Doped Turbo* Codes (ten Brink)
- *Concatenated Tree* Codes (Ping and Wu)

We Should Include Variations On the Turbo-Theme ("Turbolike" Codes)

- *Gallager* (Low-Density Parity-Check) Codes
- *Irregular LDPC* Codes (Luby, Mitzenmacher, Richardson, Shokrollahi, Speilman, Stemann, and Urbanke)
- *Repeat-Accumulate* Codes (Divsalar, Jin, McEliece)
- *Irregular Turbo* Codes (Frey and MacKay)
- *Asymmetric Turbo* Codes (Costello and Massey)
- *Mixture Turbo* Codes (Divsalar, Dolinar, and Pollara)
- *Doped Turbo* Codes (ten Brink)
- *Concatenated Tree* Codes (Ping and Wu)

$\vdots$

Turbolike Codes Have Certainly Met the Challenge on the Binary Erasure Channel !



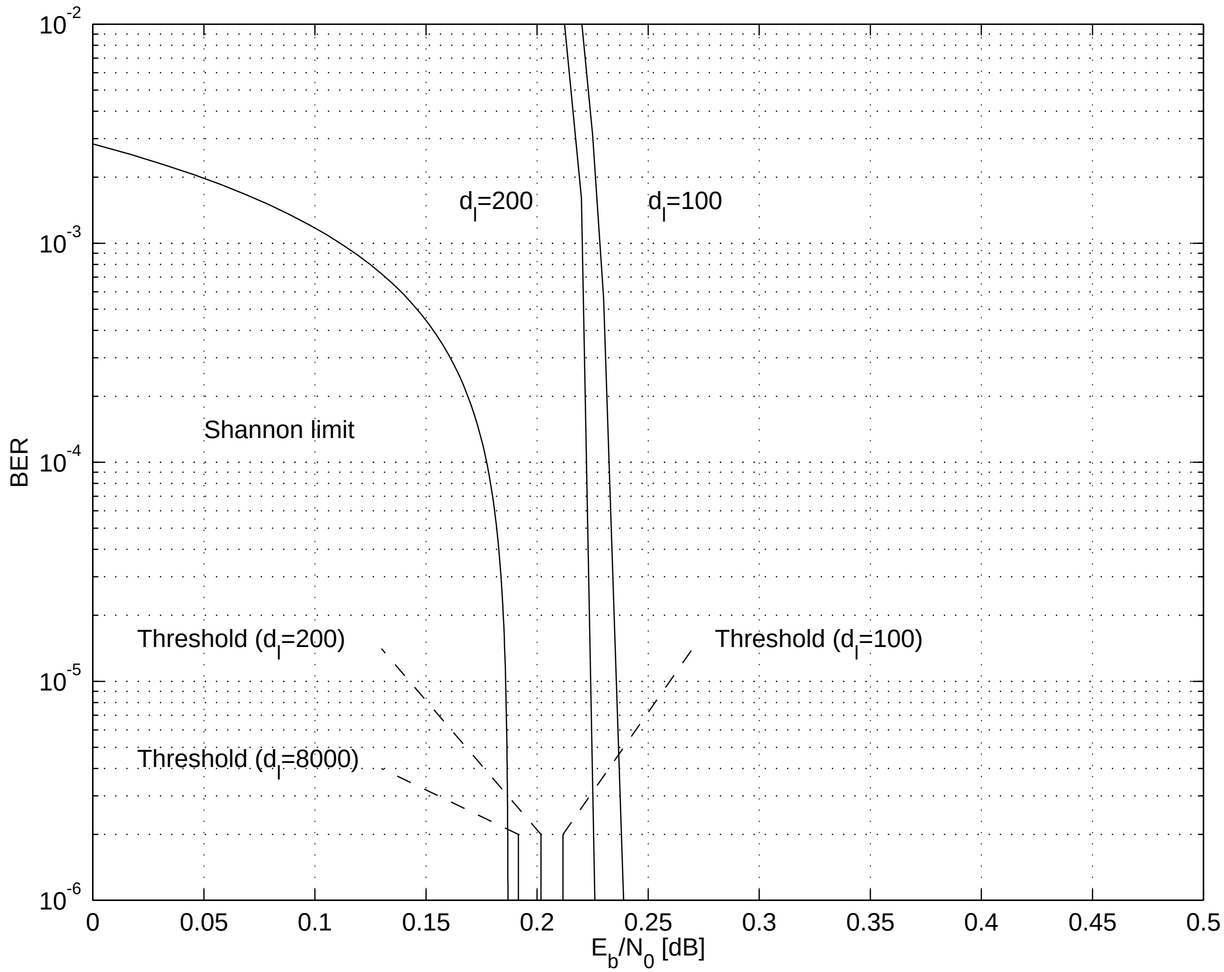
Theorem: *For the binary erasure channel, for the ensemble of (degree-profile optimized) irregular LDPC codes with iterative belief propagation decoding, as* $\epsilon \to 0$,

$$\chi_D(\epsilon, p) = O(\log \frac{1}{\epsilon})$$

Irregular LDPC Codes, Density Evolution

(Luby, Mitzenmacher, Shokrollahi, Speilman, Stemann)

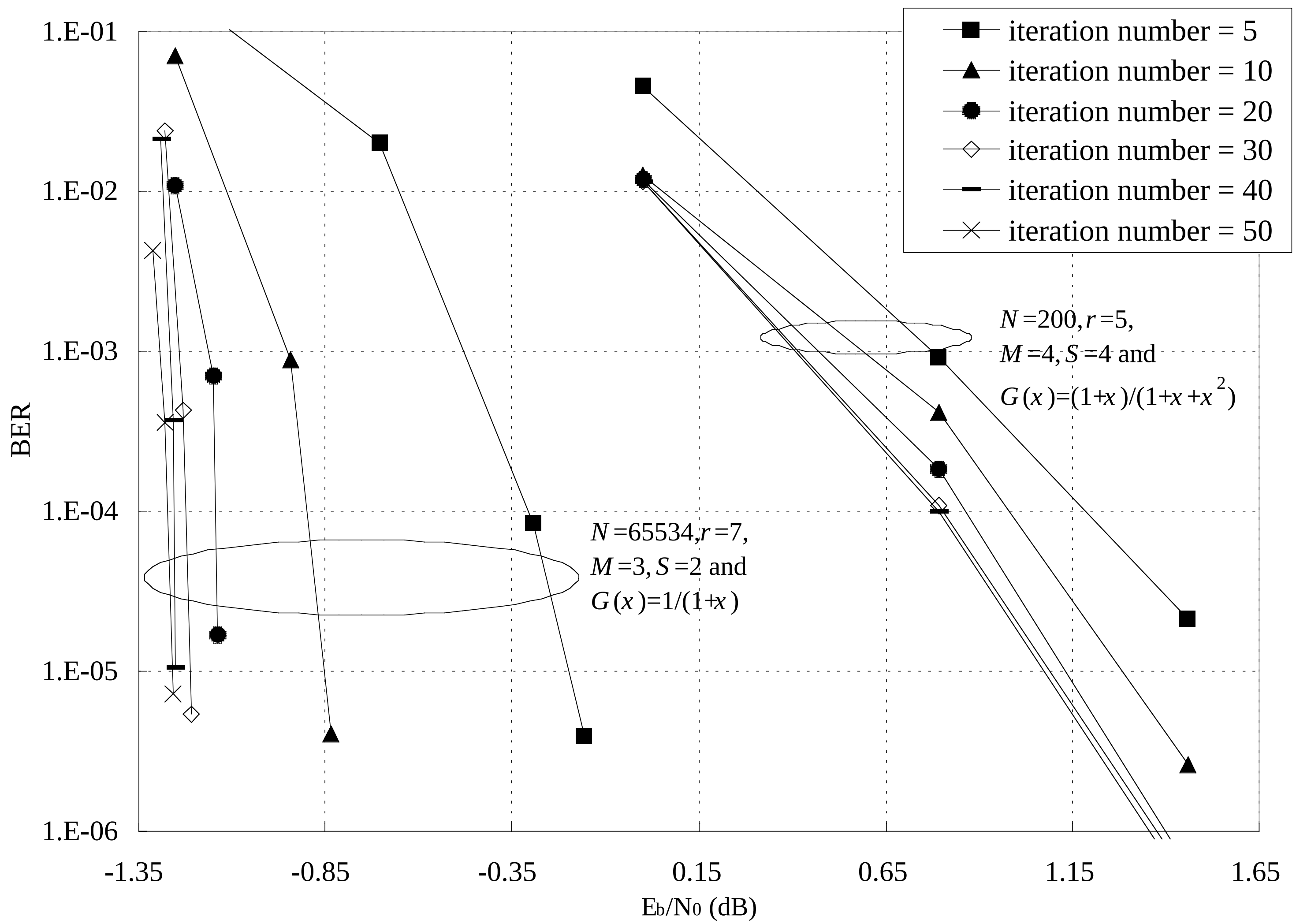
Turbolike Codes Appear to Have Met the Challenge on the Additive Gaussian Noise Channel (I)



Irregular LDPC Codes, Density Evolution

(Chung, Forney, Richardson, Urbanke, 2001)

Turbolike Codes Appear to Have Met the Challenge on the Additive Gaussian Noise Channel (II)



Turbo-Hadamard Codes (Ping, Leung, Wu, ISIT 2001)

A Conjecture

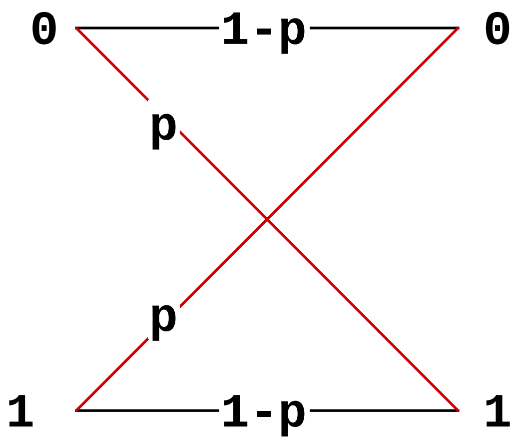
Turbolike codes meet the Shannon challenge for any symmetric binary input channel. To be precise: there exists a sequence of turbolike code ensembles plus matched iterative belief propagation decoding algorithms, such that for any fixed p, as $\epsilon \to 0$,

$$\chi_E(\epsilon, p) = O(\log \frac{1}{\epsilon})$$

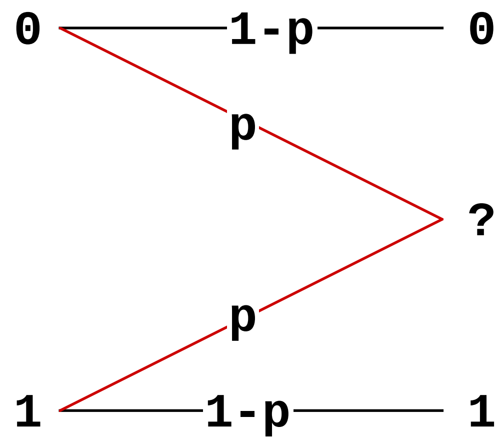
$$\chi_D(\epsilon, p) = O(\frac{1}{\epsilon} \log \frac{1}{\epsilon})$$

(Khandekar and McEliece, ISIT 2001)

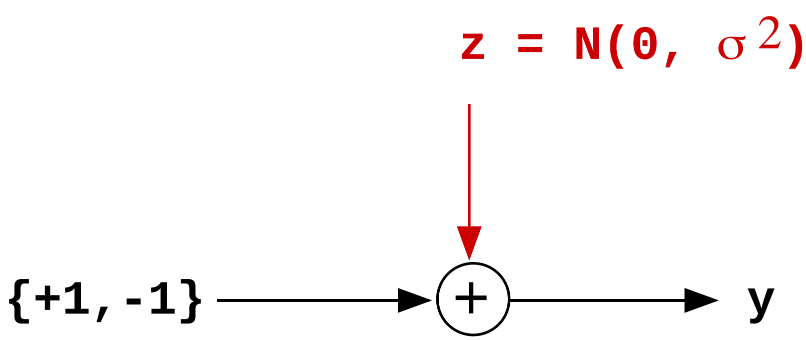
Three Garden-Variety SBIC's



Binary Symmetric

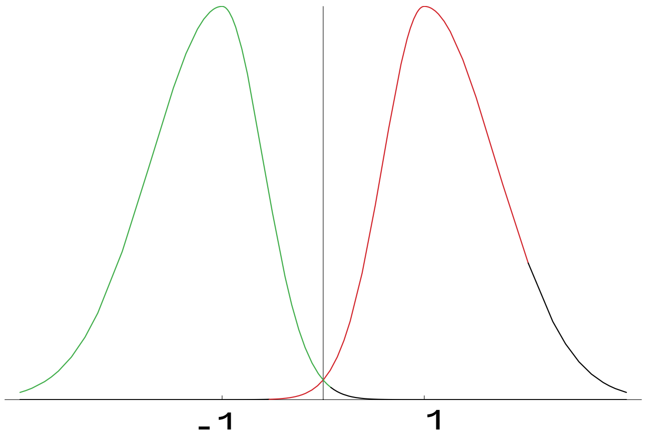


Binary Erasure



Binary Input, Additive Gaussian Noise

The Generalization (Gallager, 1963)



Definition: *A* *symmetric binary-input channel* *is a memoryless, discrete-time channel with*

- Input alphabet $X = \{+1, -1\}$.
- Output alphabet $Y \subseteq$ Real Numbers.
- Transition probabilities

$$p(y|x = +1) = f(y)$$
$$p(y|x = -1) = f(-y).$$

Examples of SBIC's

- The Binary Erasure Channel:

$$f(y) = (1-p)\delta(y-1) + p\delta(y).$$

- The Binary Symmetric Channel:

$$f(y) = (1-p)\delta(y-1) + p\delta(y+1).$$

- Additive Gaussian Noise:

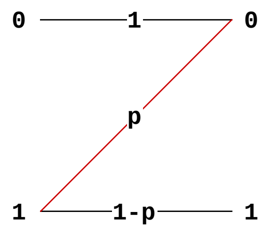
$$f(y) = K\exp((y-1)^2/2\sigma^2).$$

- Fast Rayleigh Fading (noncoherent model):

$$f(y) = \begin{cases} K\exp(-y/A) & \text{if } y \geq 0 \\ K\exp(y(1+A)/y) & \text{if } y < 0. \end{cases}$$

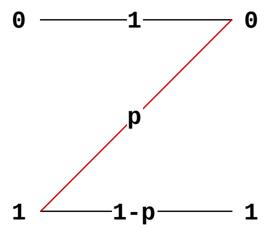
But What About Non-SBIC's, i.e., (Memoryless) Channels that are

But What About Non-SBIC's, i.e., (Memoryless) Channels that are

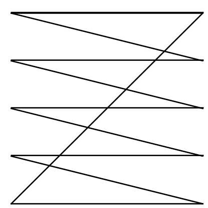


Nonsymmetric?

But What About Non-SBIC's, i.e., (Memoryless) Channels that are

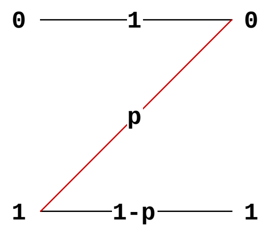


Nonsymmetric?

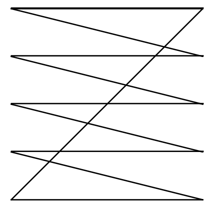


Nonbinary?

But What About Non-SBIC's, i.e., (Memoryless) Channels that are



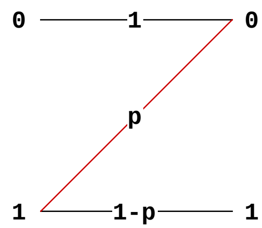
Nonsymmetric?



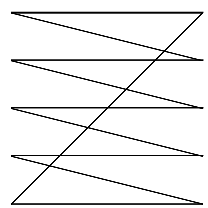
Nonbinary?

Multiuser?

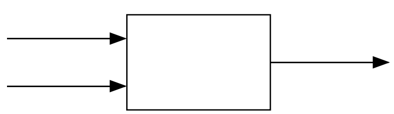
But What About Non-SBIC's, i.e., (Memoryless) Channels that are



Nonsymmetric?



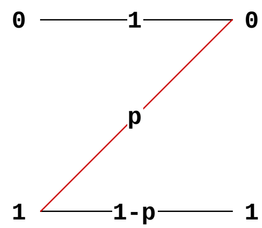
Nonbinary?



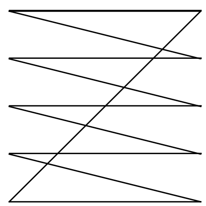
Multiuser?

Etc.?

But What About Non-SBIC's, i.e., (Memoryless) Channels that are



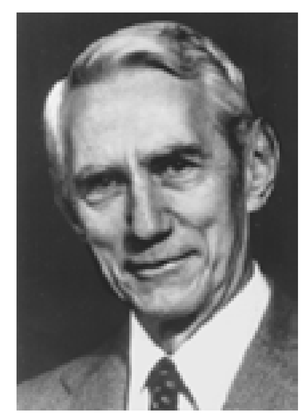
Nonsymmetric?



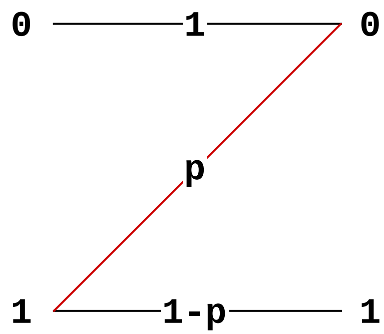
Nonbinary?

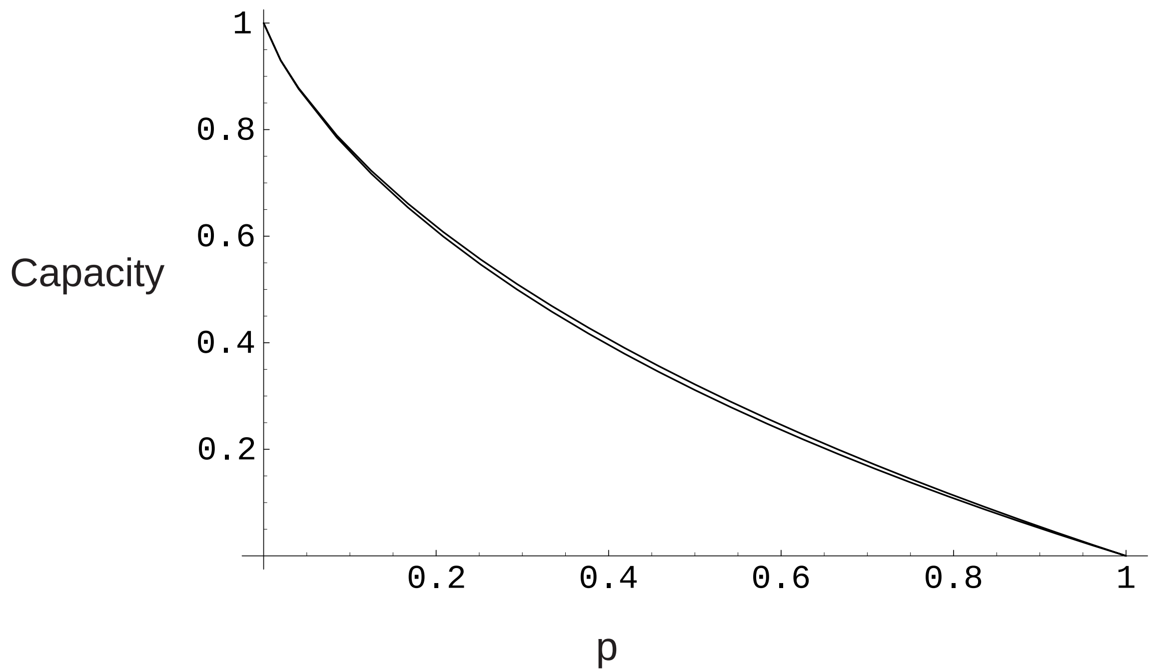
Multiuser?

Etc.?

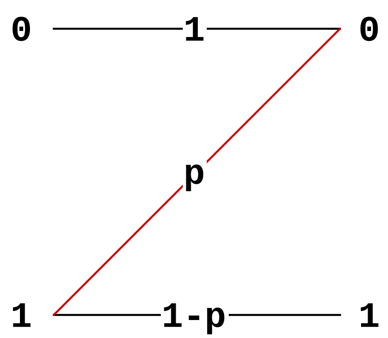


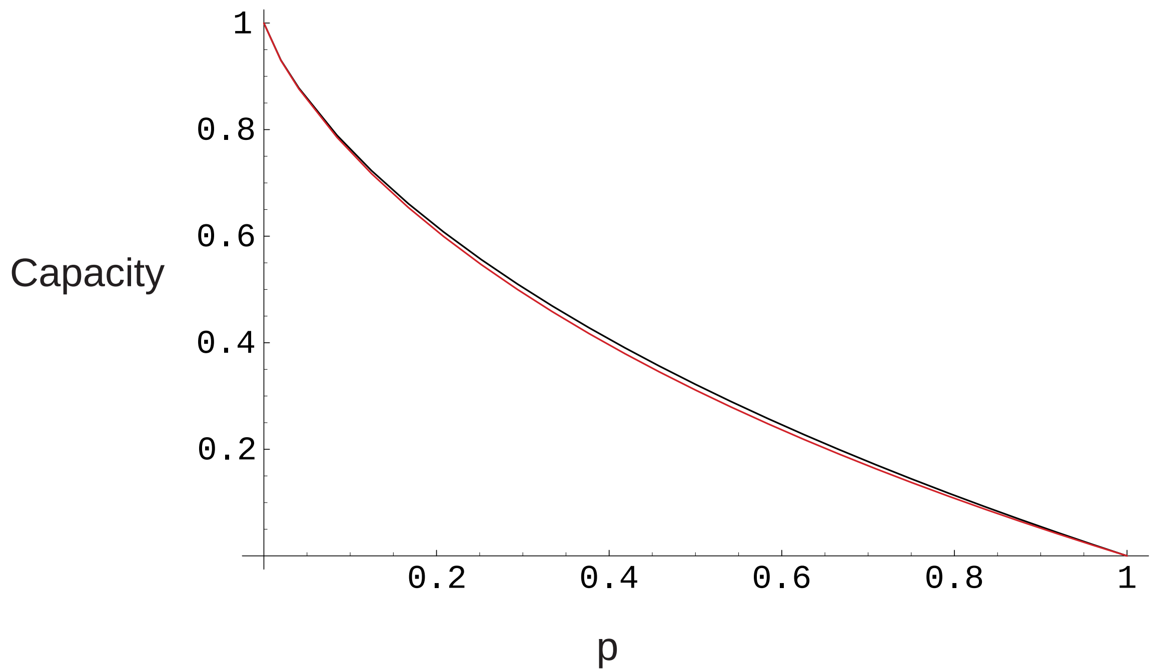
The Simplest Nonsymmetric Channel: The Z-Channel



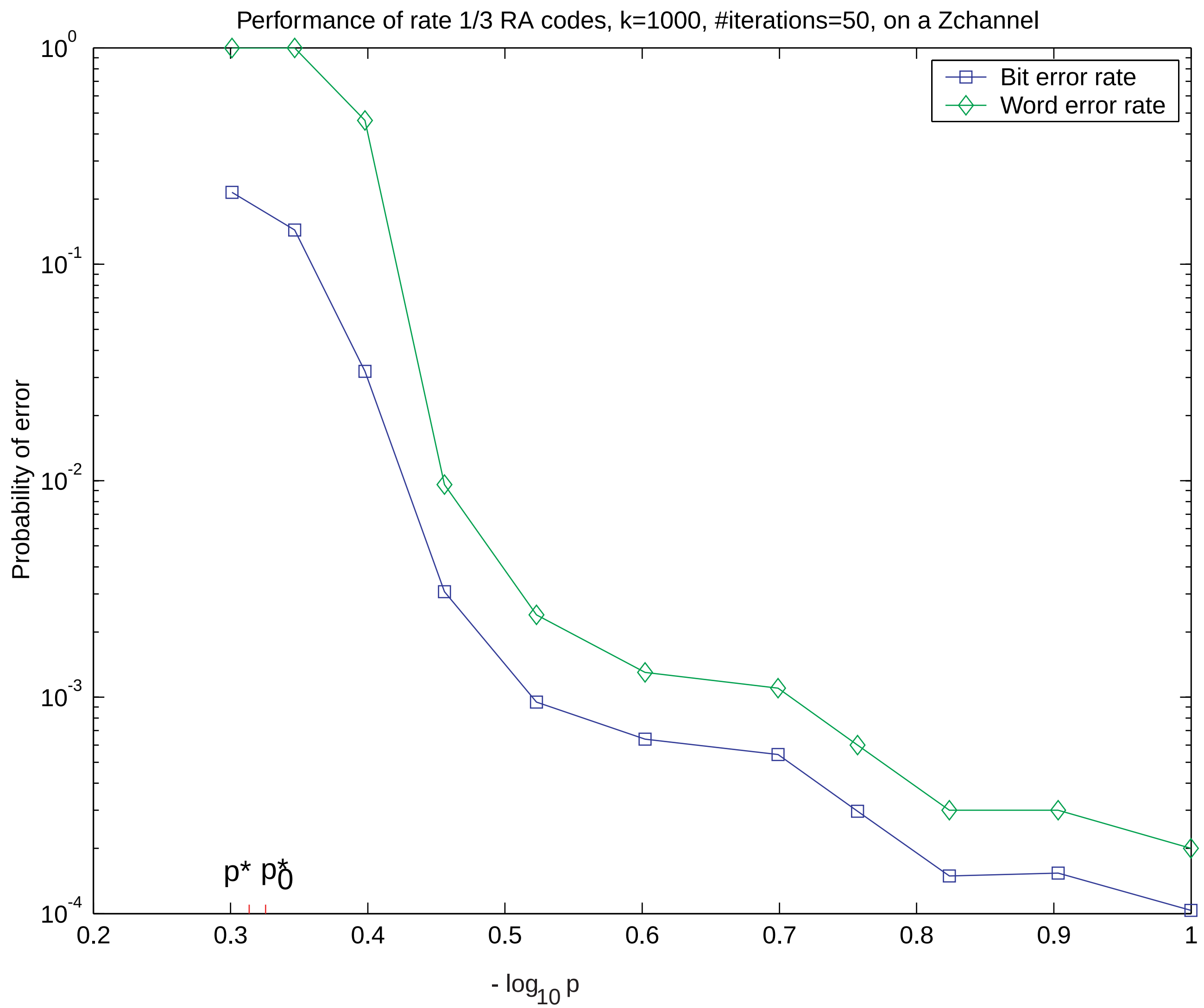


The Simplest Nonsymmetric Channel: The Z-Channel





An Experiment on the Z-channel
(Rate 1/3 Repeat-Accumulate Code, $k = 1000$.)

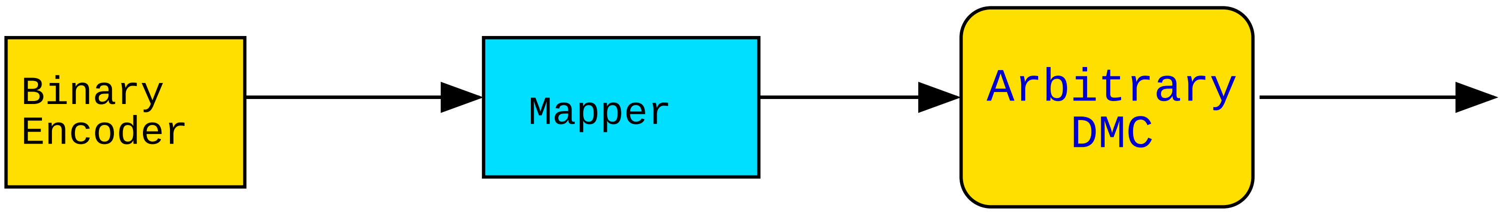


Can We Deal More Honestly With the Problem?

An Old Theorem of Gallager (1968)

Theorem. *Binary linear codes can be used to achieve capacity on an arbitrary discrete memoryless channel.*

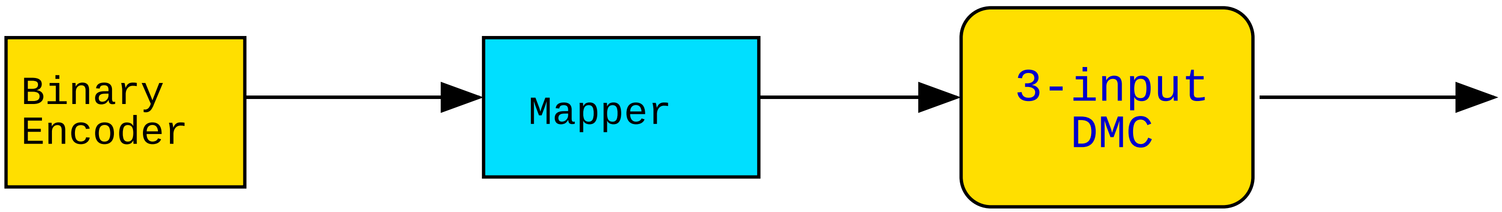
Proof: Encoding algorithm:



An Old Theorem of Gallager (1968)

Theorem. *Binary linear codes can be used to achieve capacity on an arbitrary discrete memoryless channel.*

Proof: Encoding algorithm:

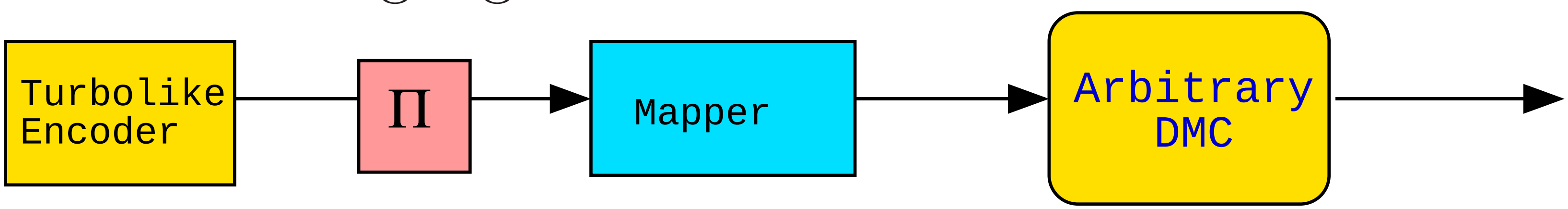



```
000
001  > a  (p = 3/8)
010

011
100  > b  (p = 3/8)
101

110
111  > c  (p = 1/4)
```

An Old Theorem of Gallager (1968)

Theorem. *Binary linear codes can be used to achieve capacity on an arbitrary discrete memoryless channel.*

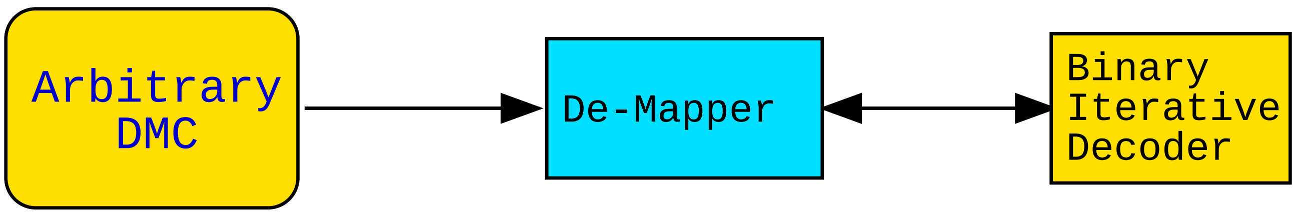
Proof: Encoding algorithm:



“Unfortunately, the problem of finding *decoding* algorithms is not so simple.” —R.G.G.

"Unfortunately, the problem of finding *decoding* algorithms is not so simple." —R.G.G.

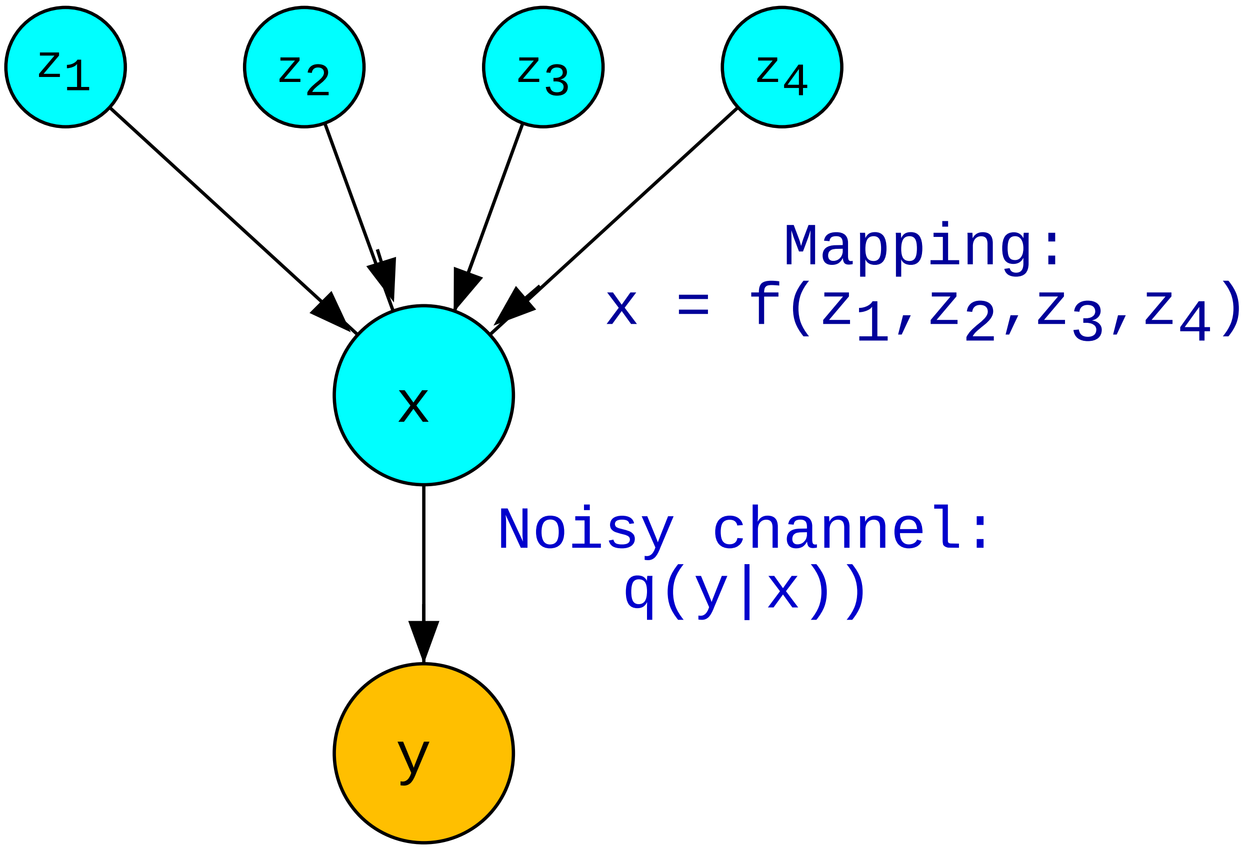
Joint Decoding-Demapping



How does the Demapper interact with the Iterative Decoder?

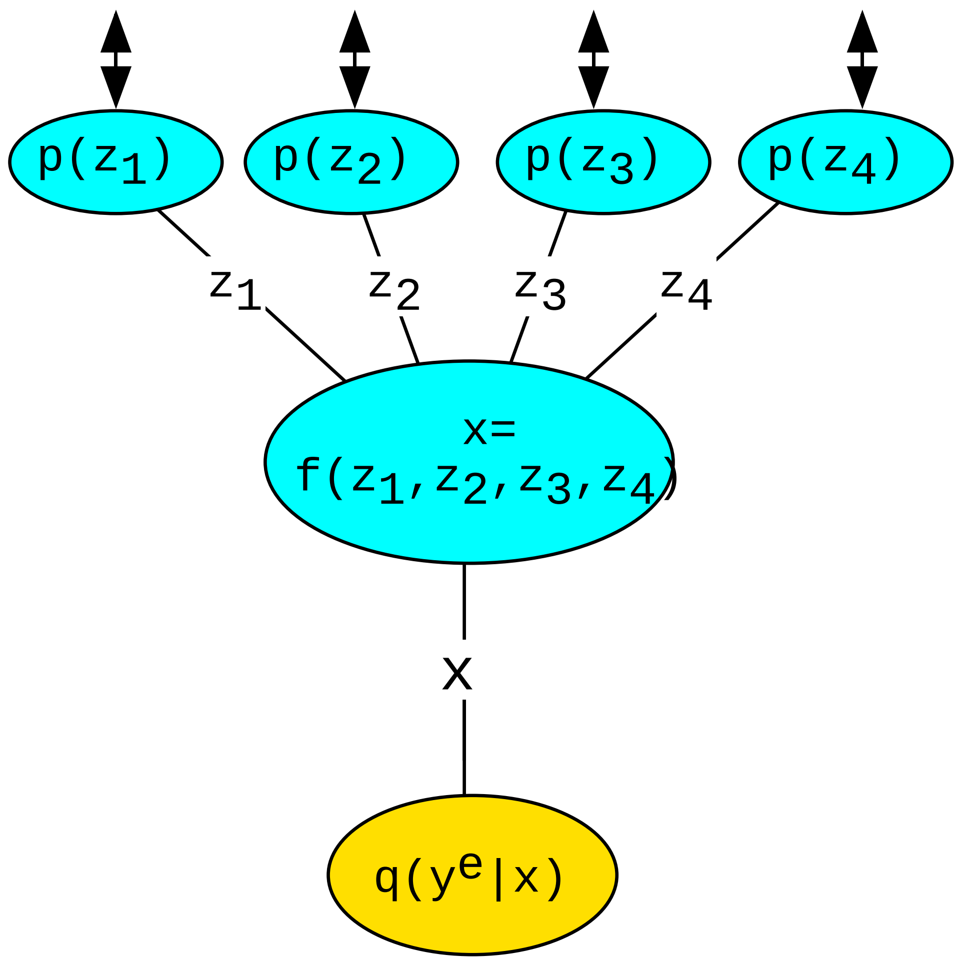
A Bayesian Network For a 4:1 Mapper

Problem: infer z_1, z_2, z_3, z_4 after observing y.

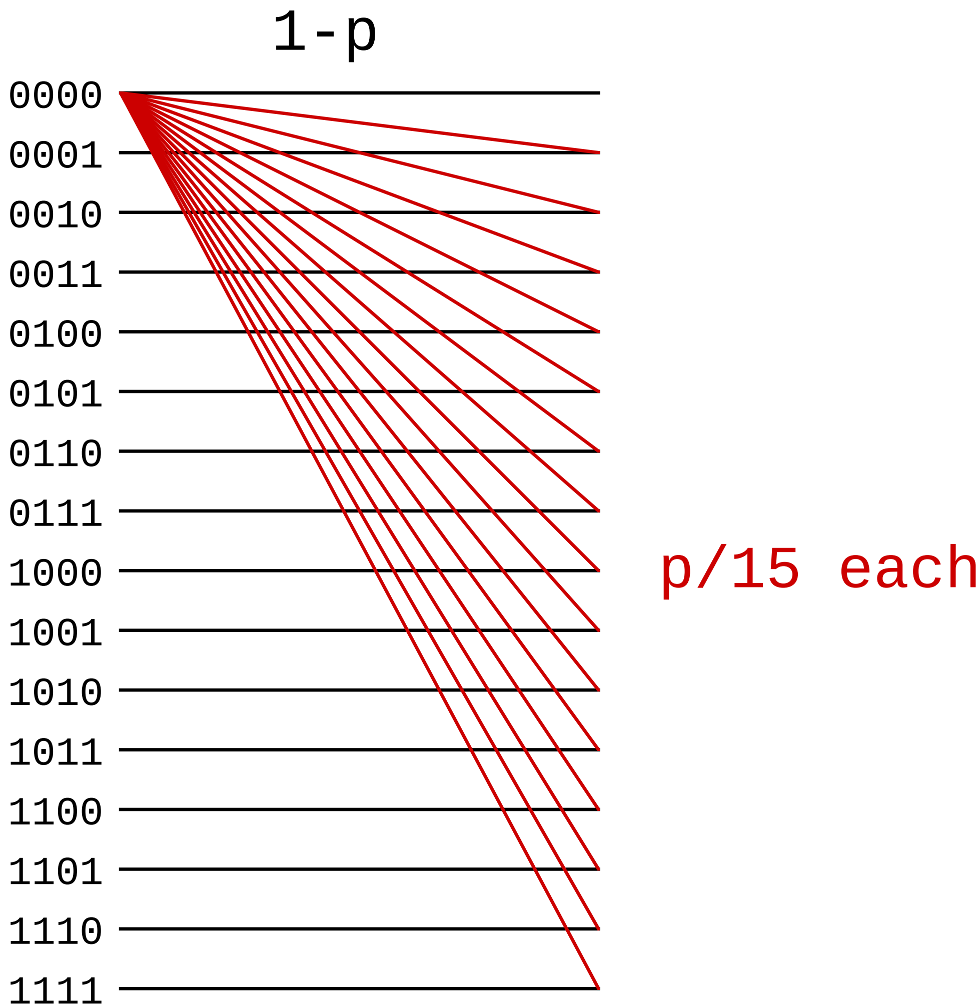


The Corresponding Junction Tree

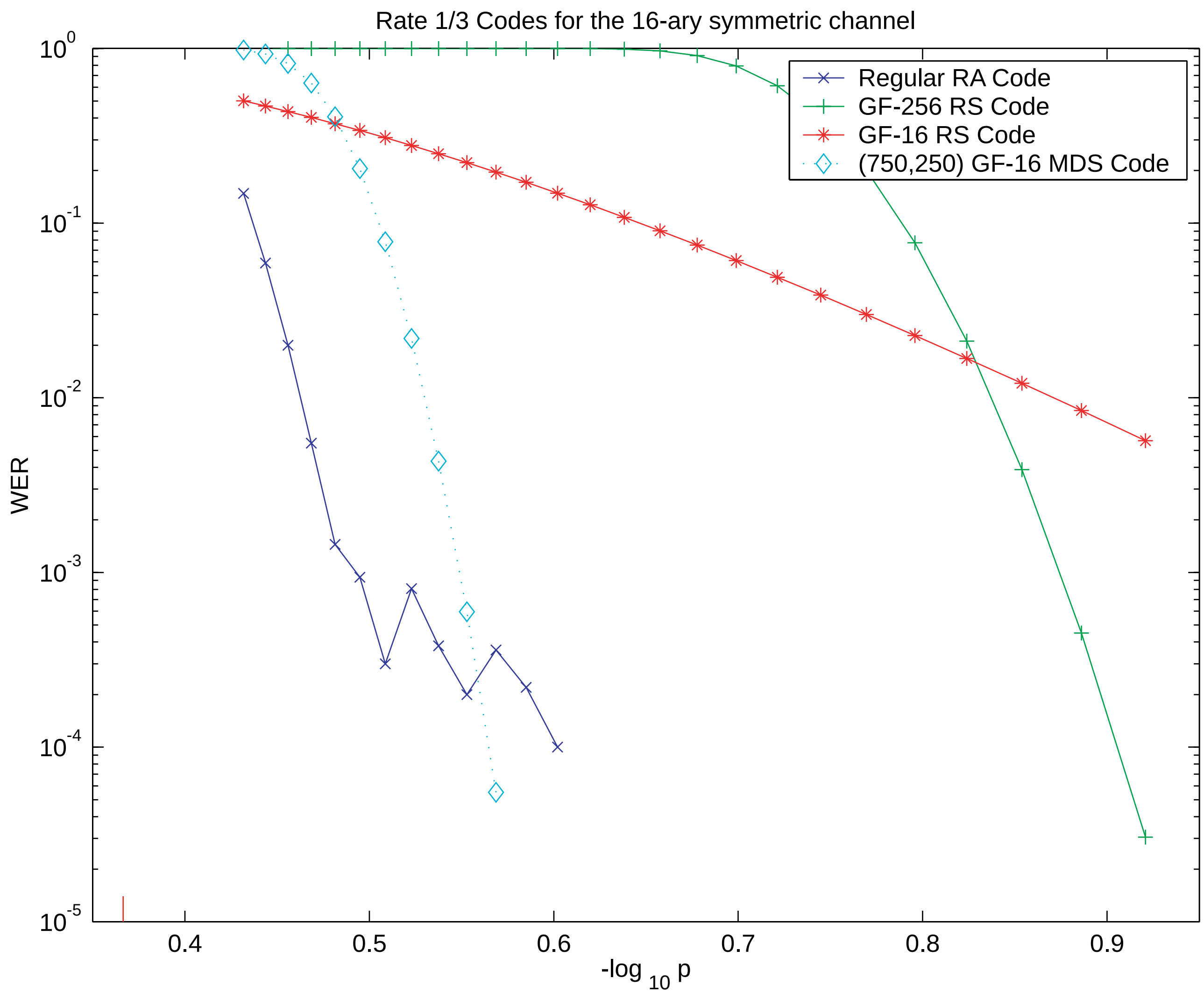
(Iterative Decoder)



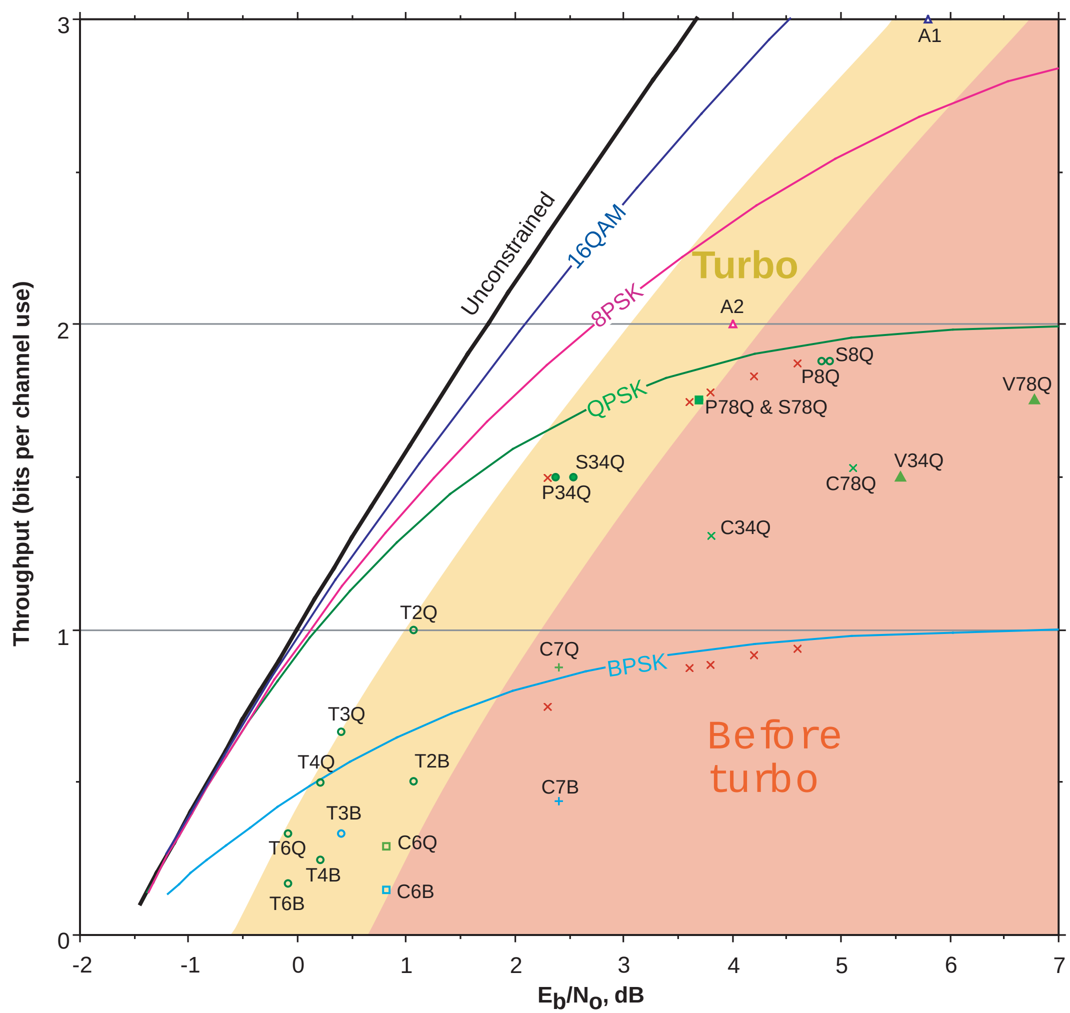
Example: The 16-ary Symmetric Channel



An Experiment on the 16-ary Symmetric Channel
$R = 1/3$ RA Code, $k = 1000$

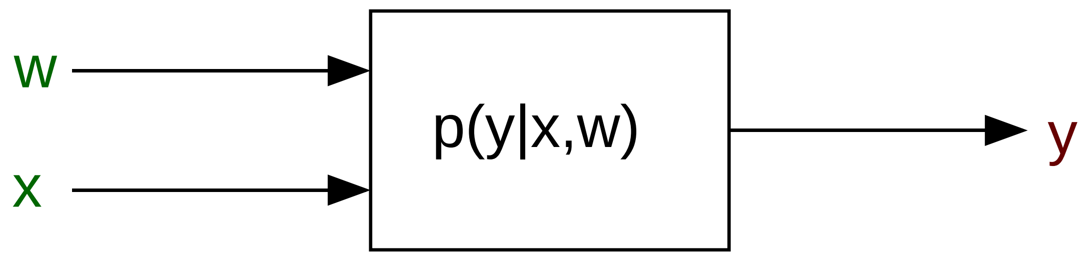


This Approach has Proved Effective on the 2D Additive White Gaussian Channel



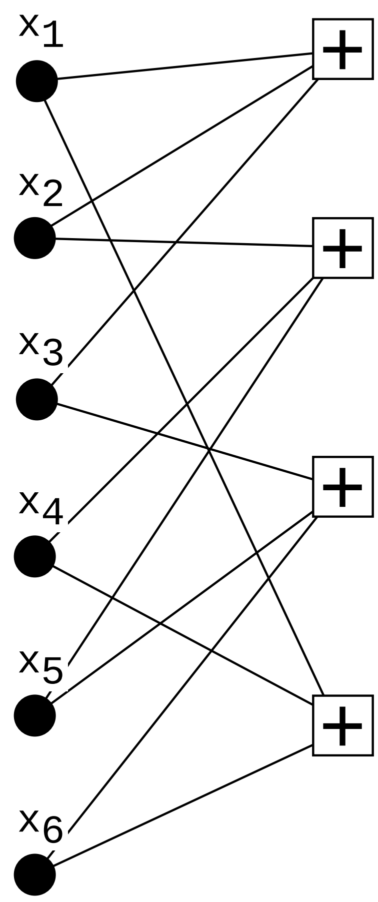
(graph due to Divsalar and Pollara)

A Simple Multiuser (Multiaccess) Channel

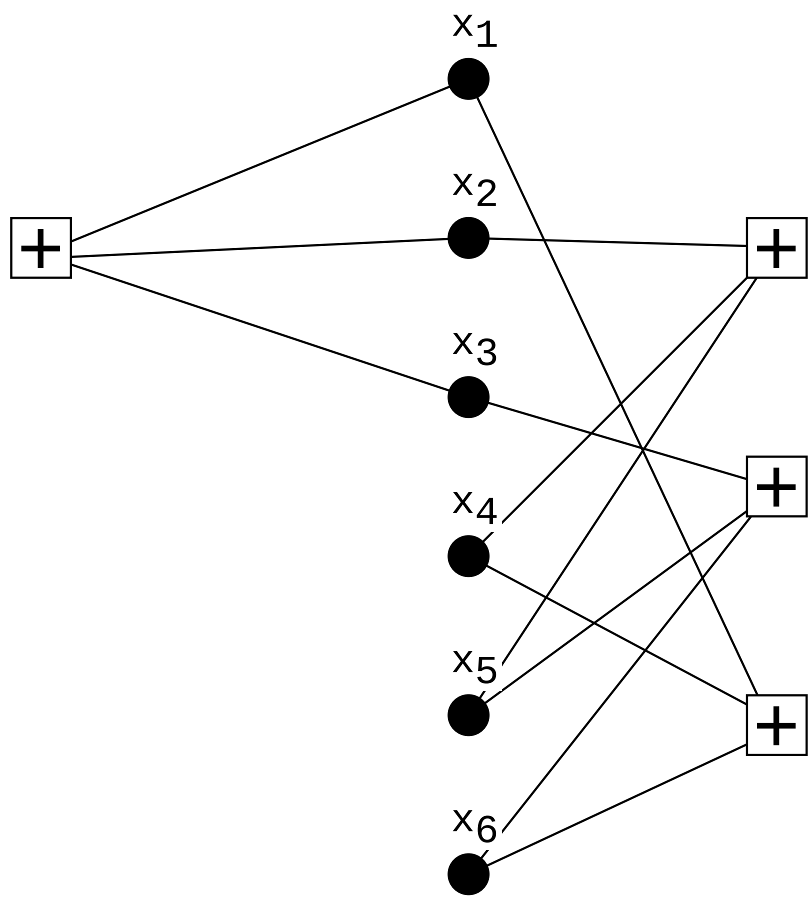


(w and x must transmit independently to y.)

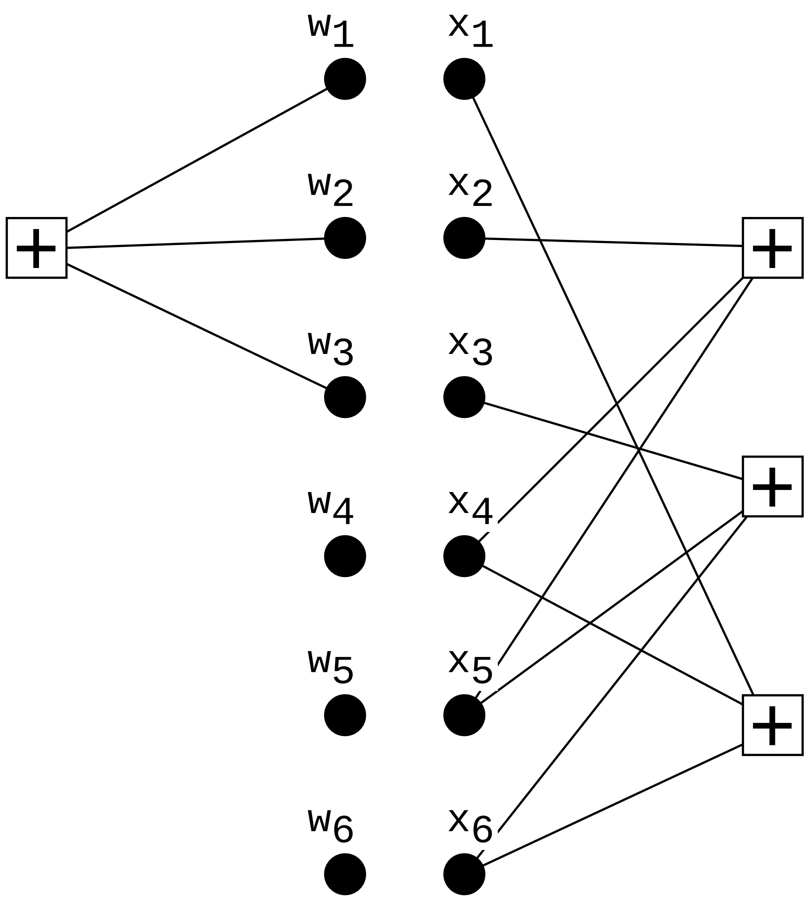
A Tanner Graph for a $(2, 3)$ LDPC Code



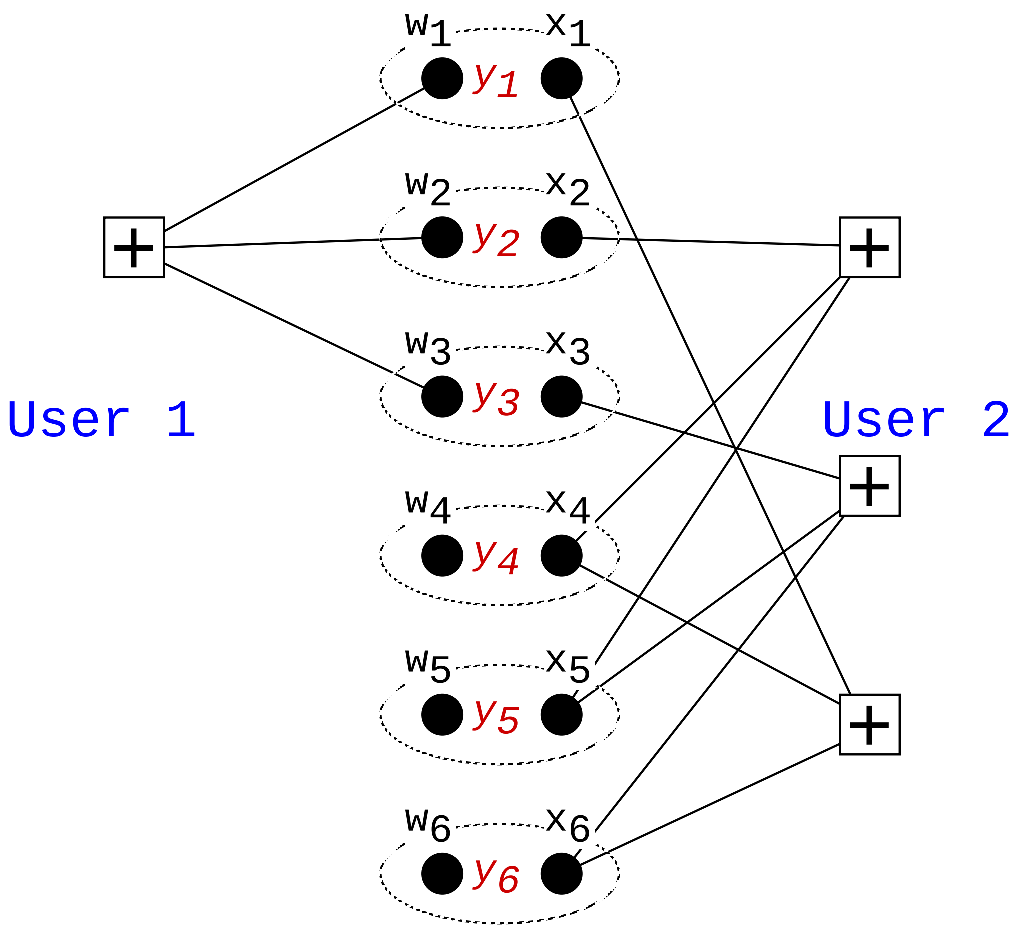
Splitting the Graph (I)



Splitting the Graph (II)

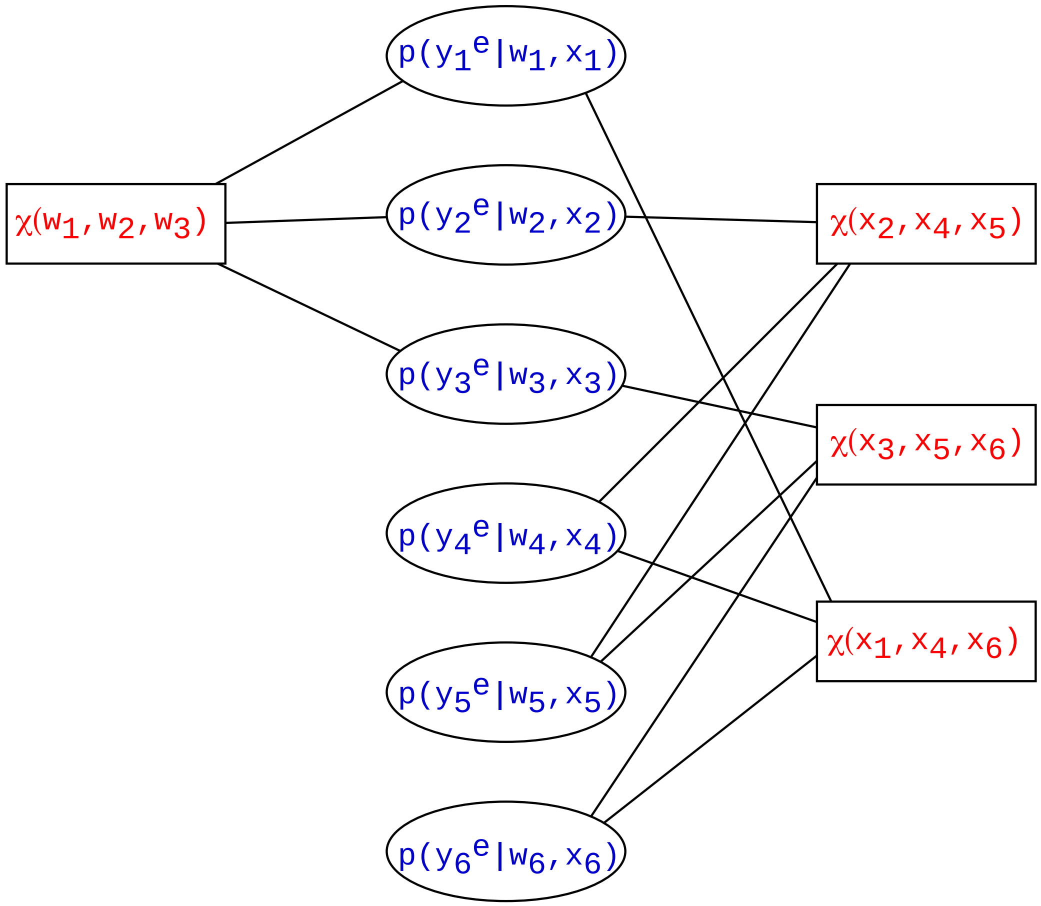


A Tanner Graph for a Multiaccess LDPC Code $(n = 6,\ R_1 = 5/6,\ R_2 = 1/2)$.



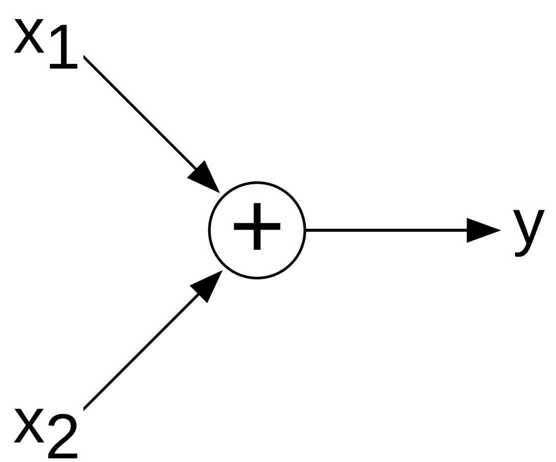
y_i = channel response to (w_i, x_i).

The Corresponding Junction Graph



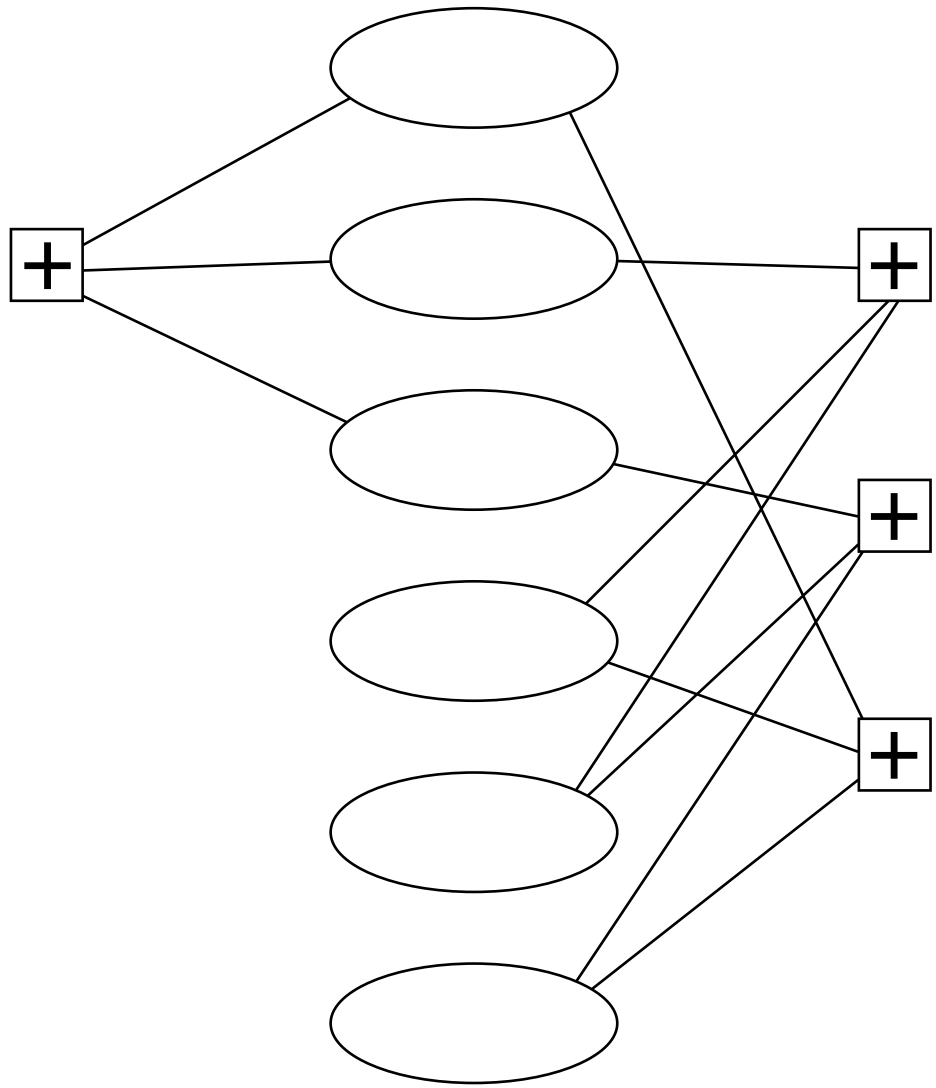
$$\chi = \begin{cases} 1 & \text{if even parity} \\ 0 & \text{if odd parity.} \end{cases}$$

Example: The Binary Adder Channel

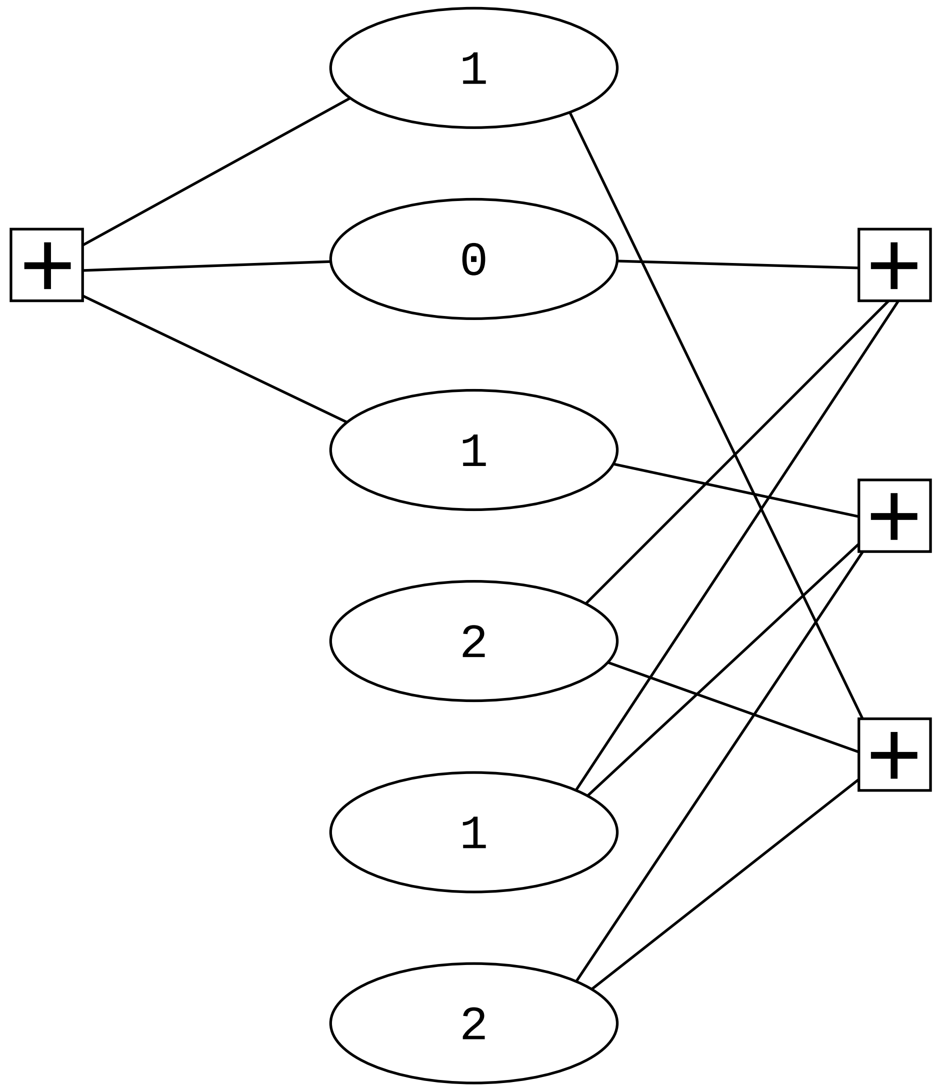


x_1	x_2	y
0	0	0
0	1	1
1	0	1
1	1	2

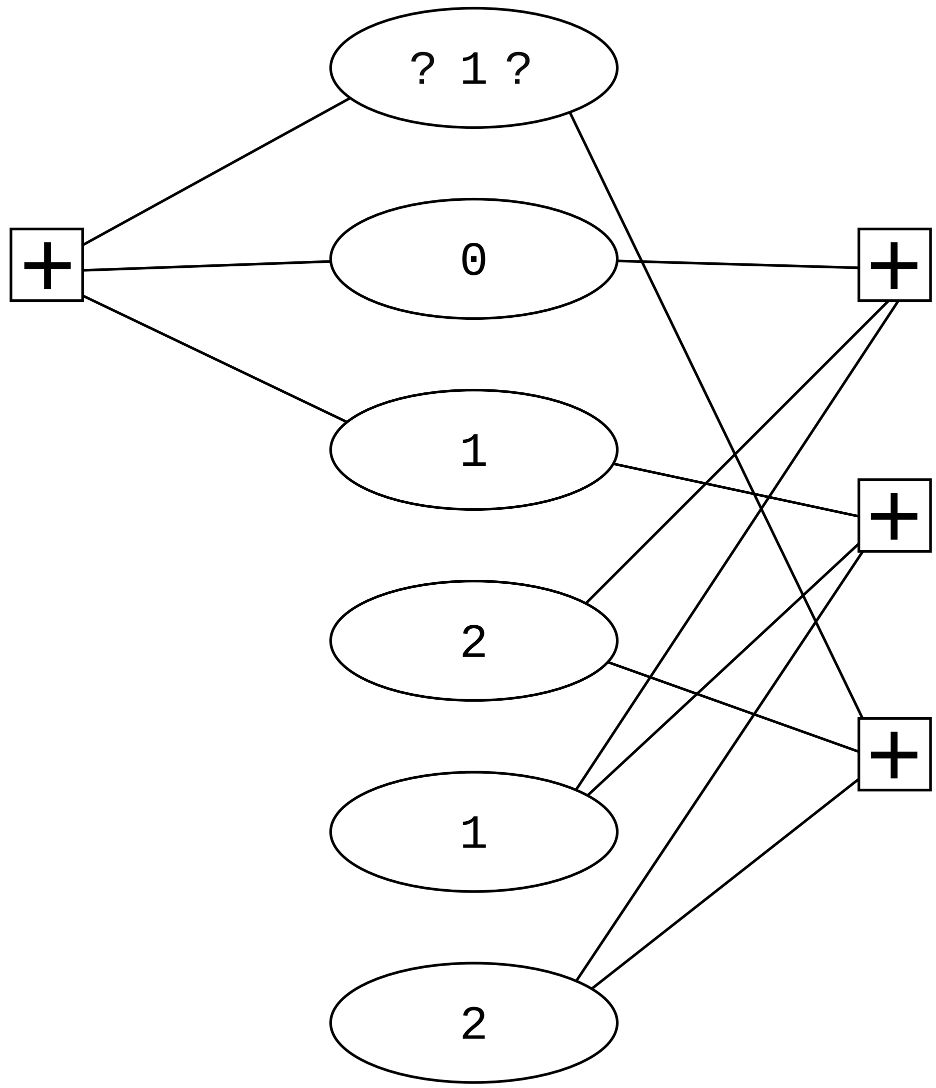
BAC Joint Decoding Example



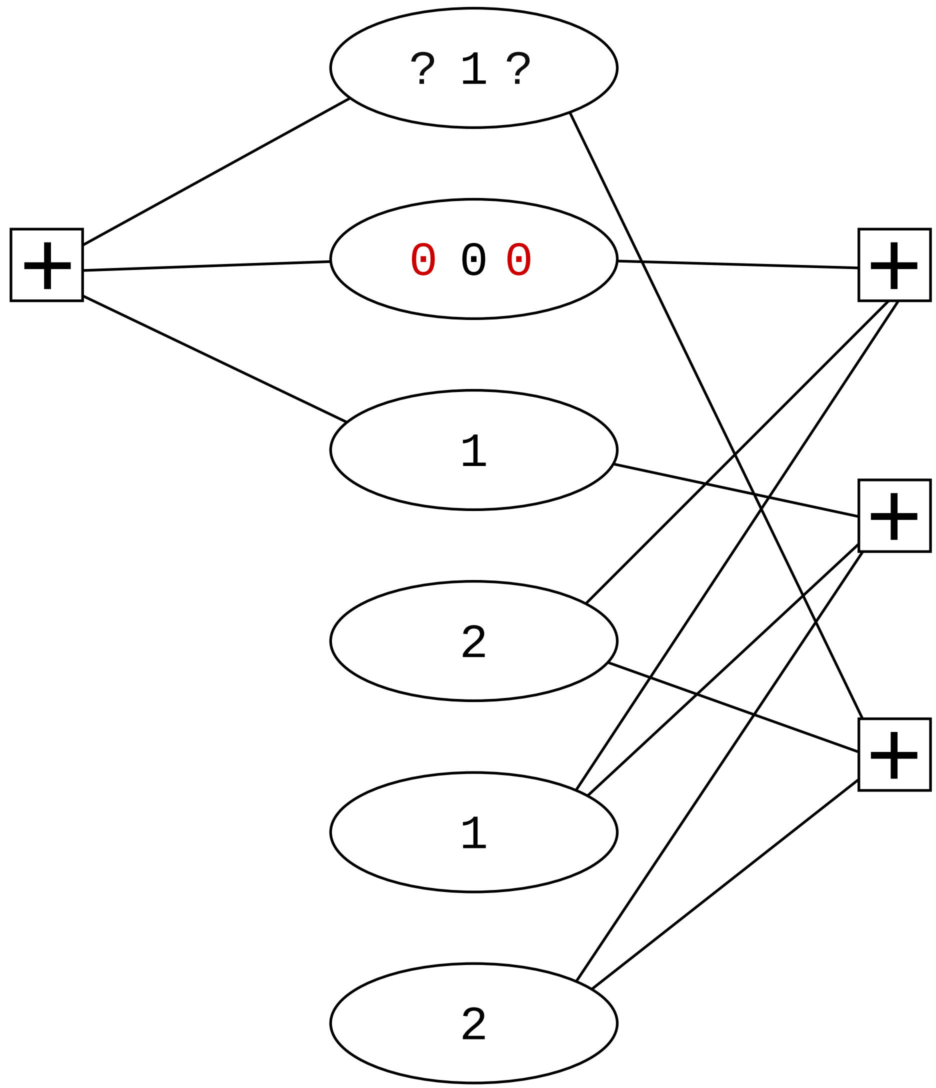
BAC Joint Decoding Example



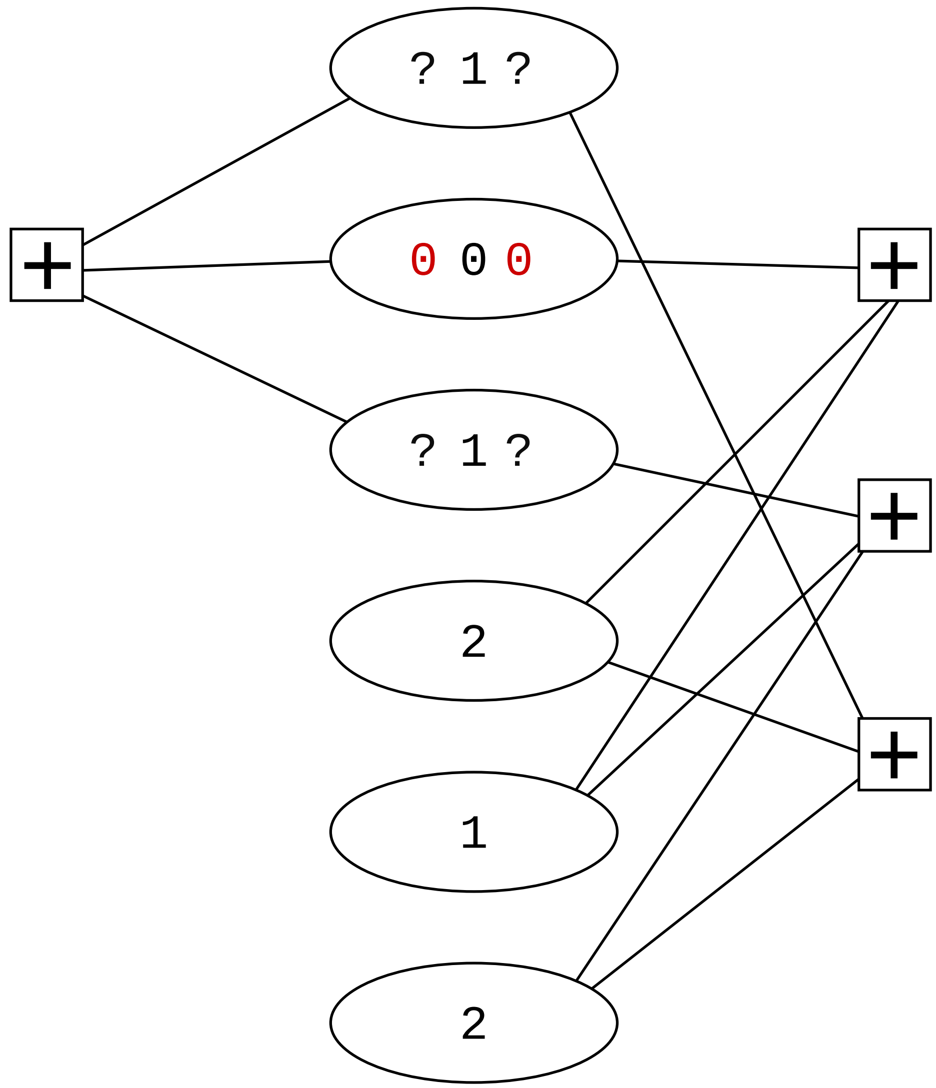
BAC Joint Decoding Example



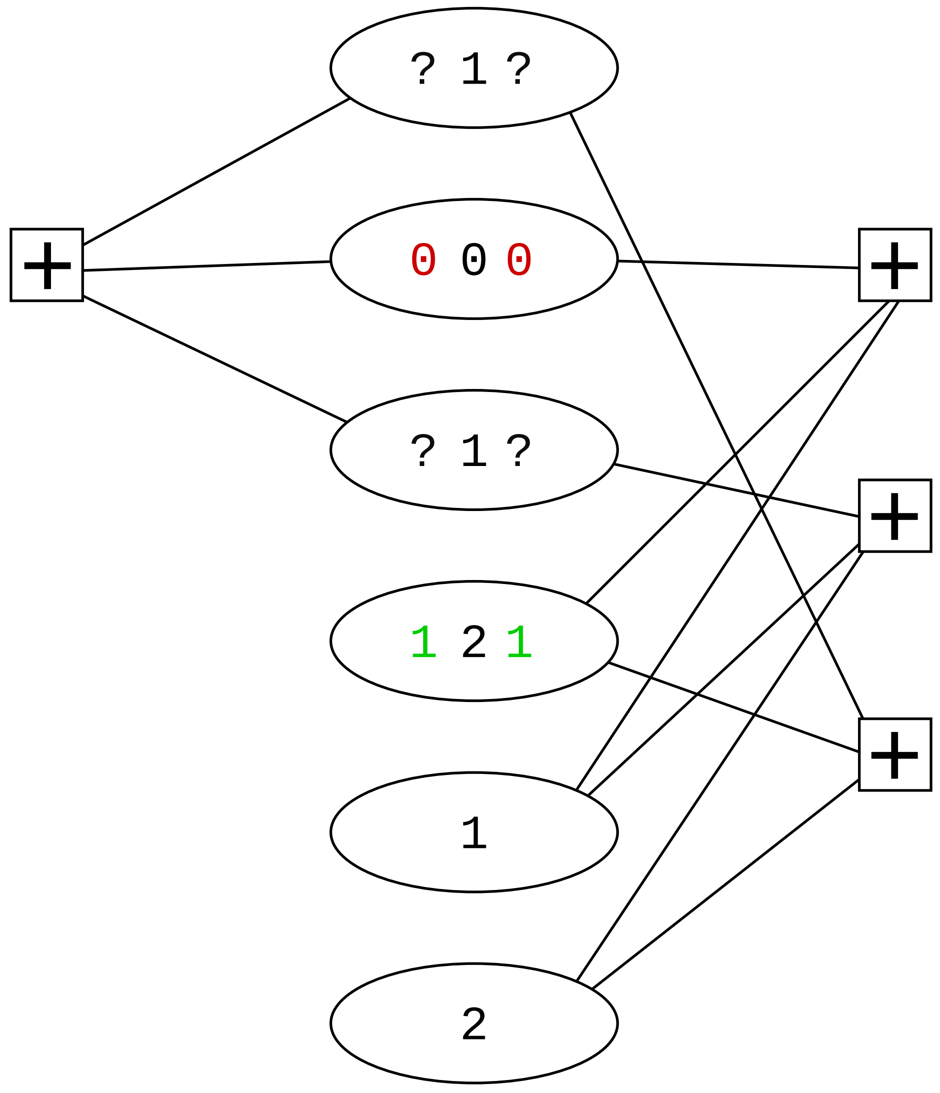
BAC Joint Decoding Example



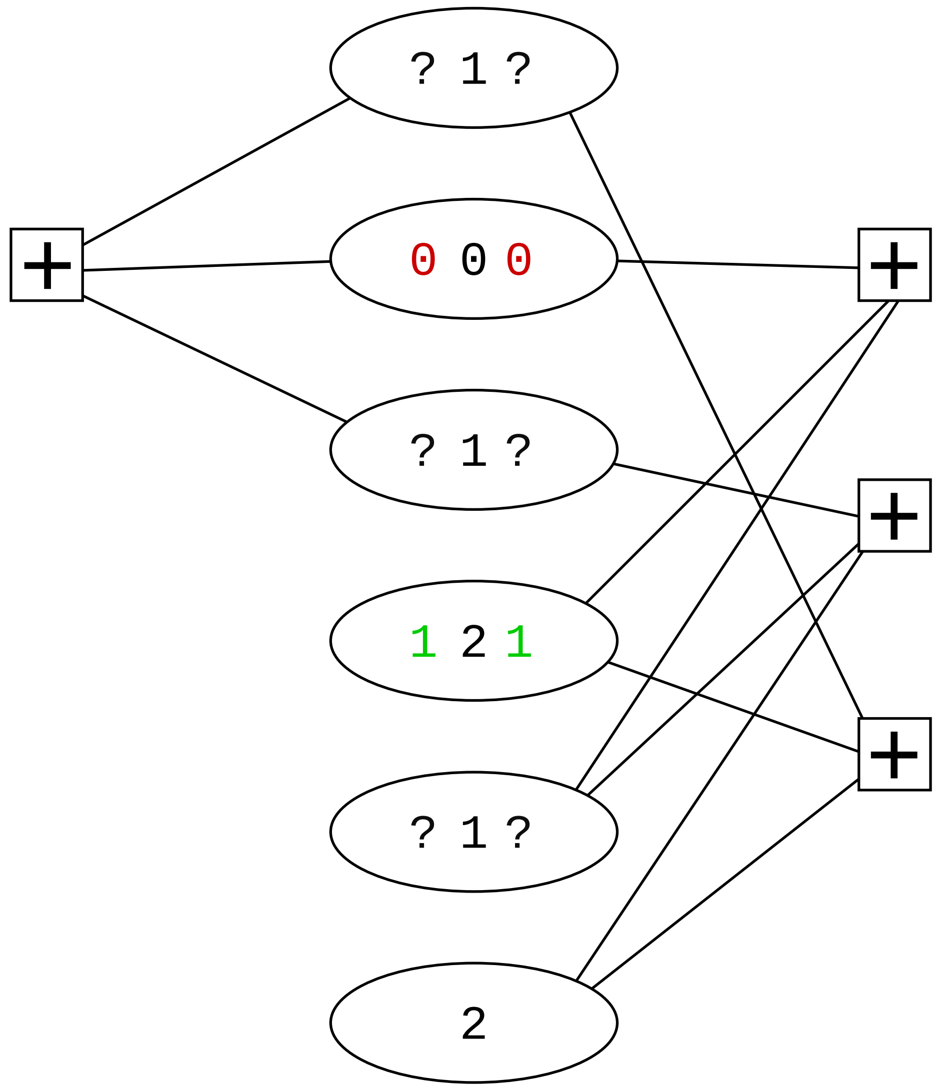
BAC Joint Decoding Example



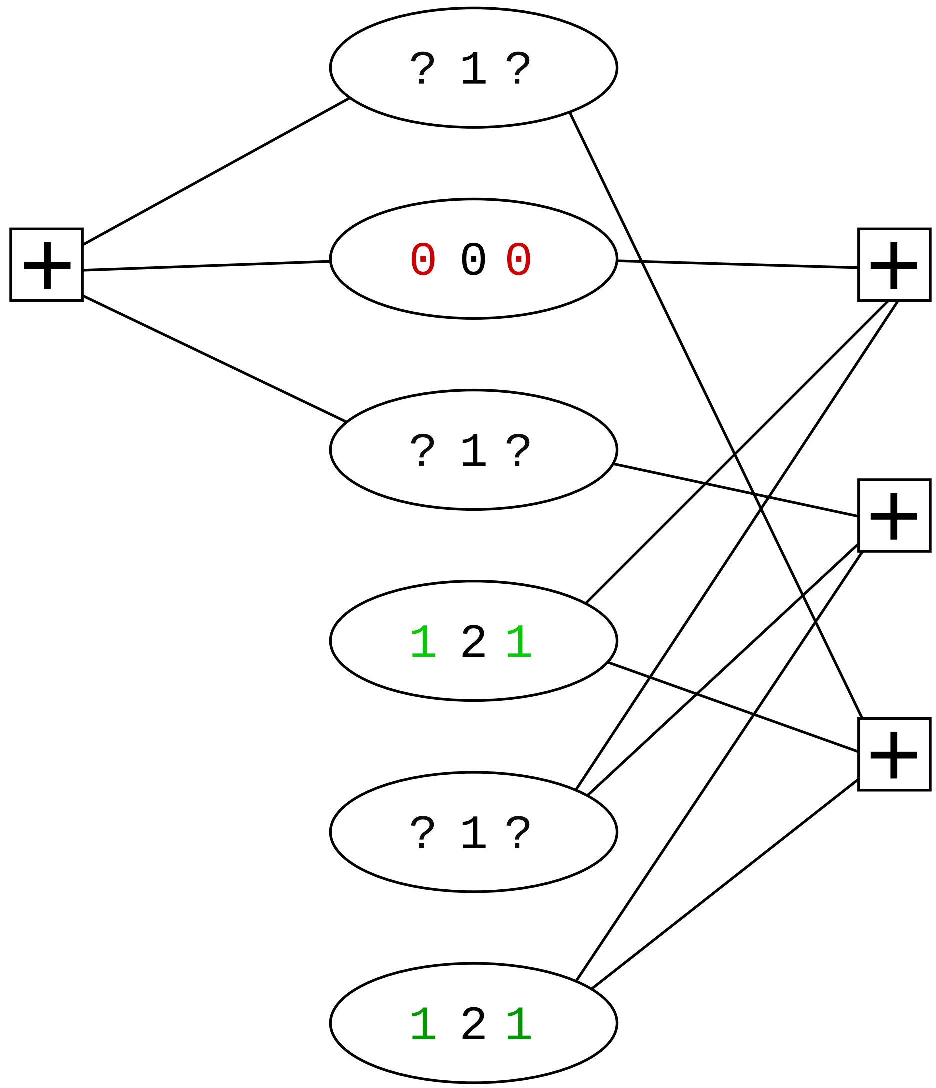
BAC Joint Decoding Example



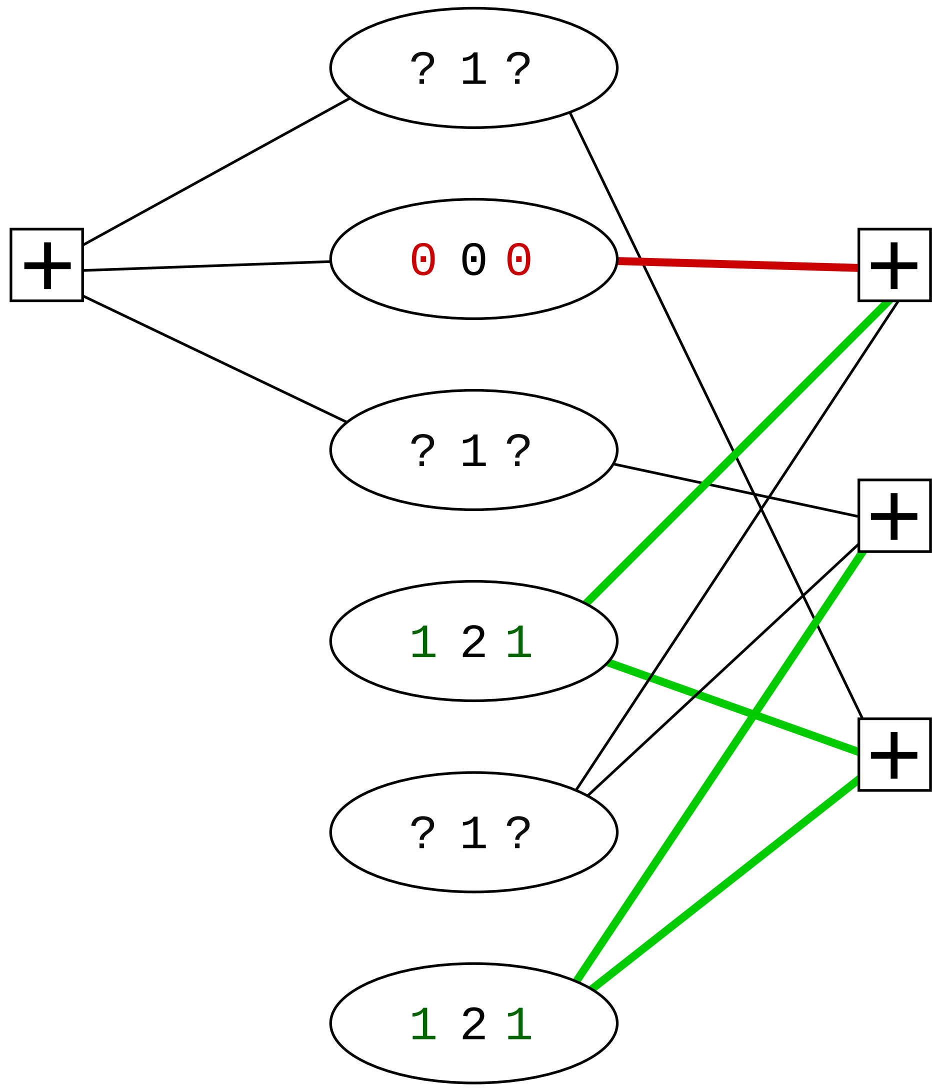
BAC Joint Decoding Example



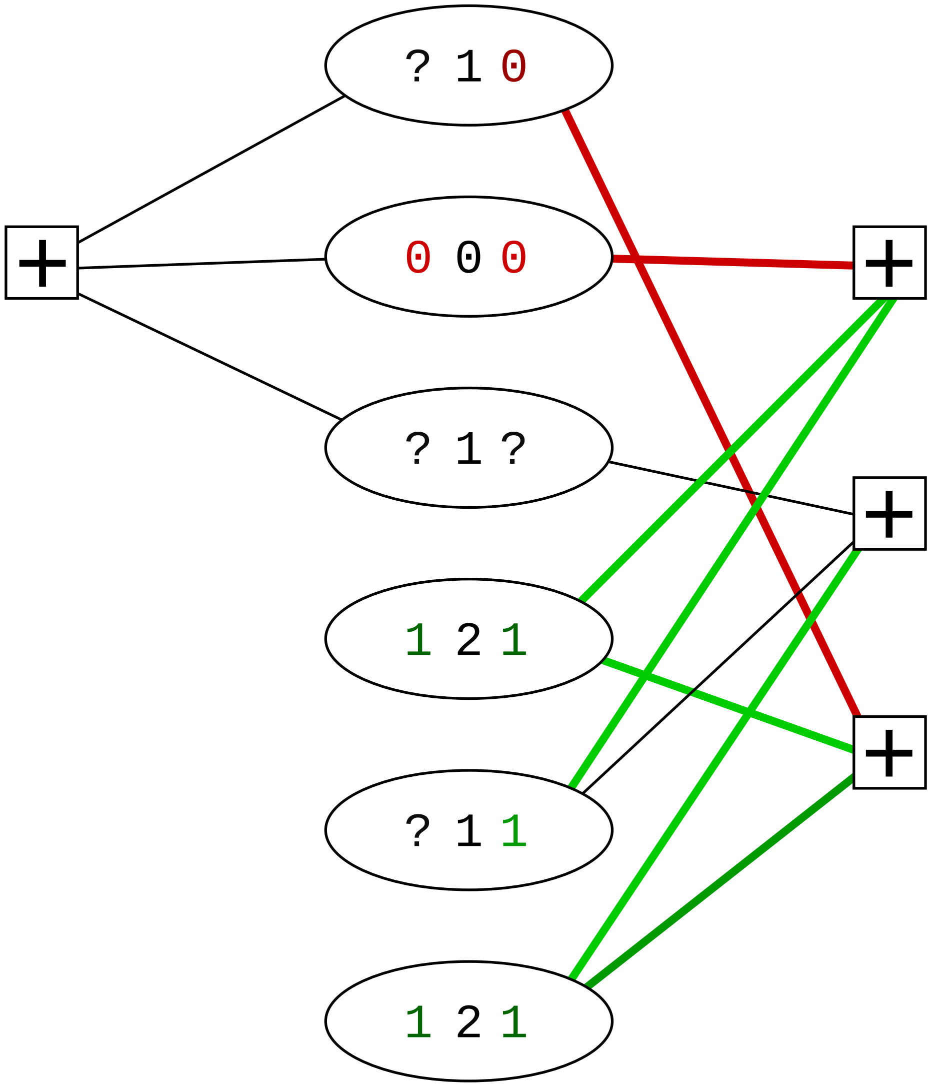
BAC Joint Decoding Example



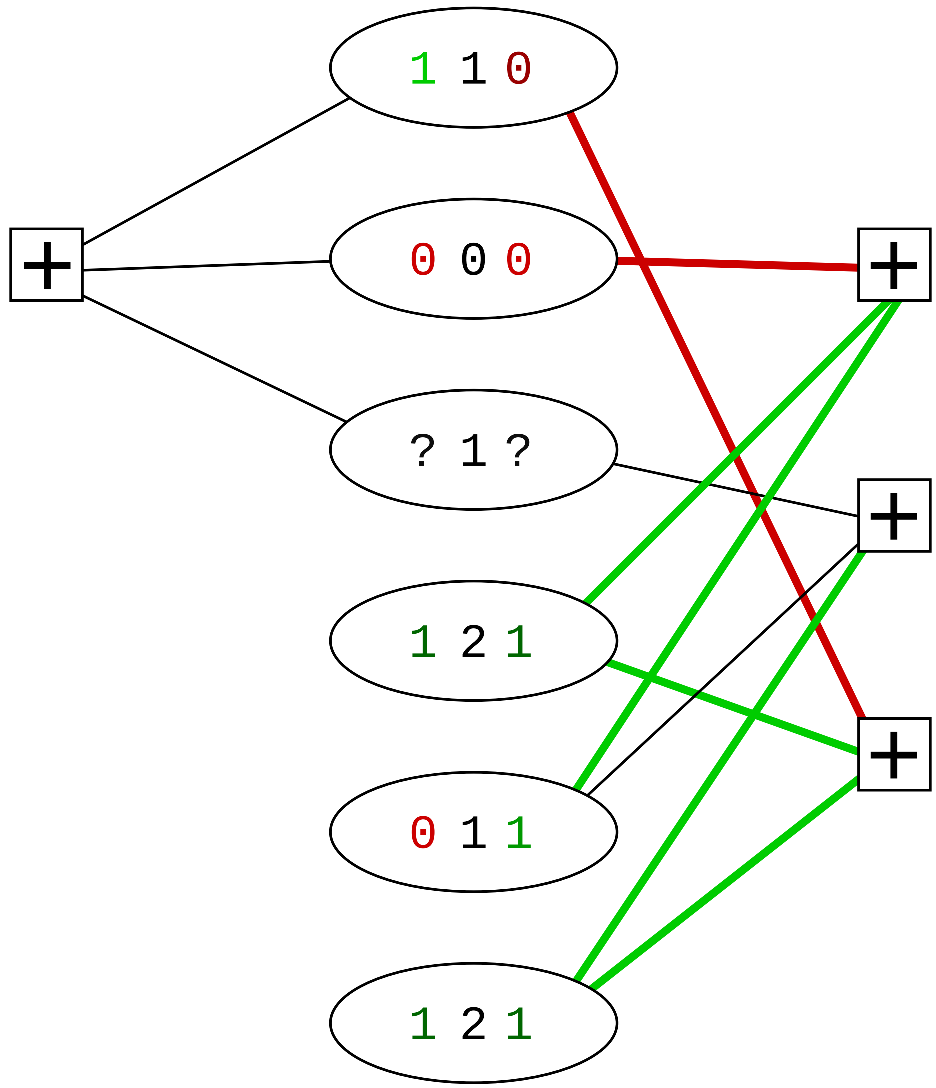
BAC Joint Decoding Example



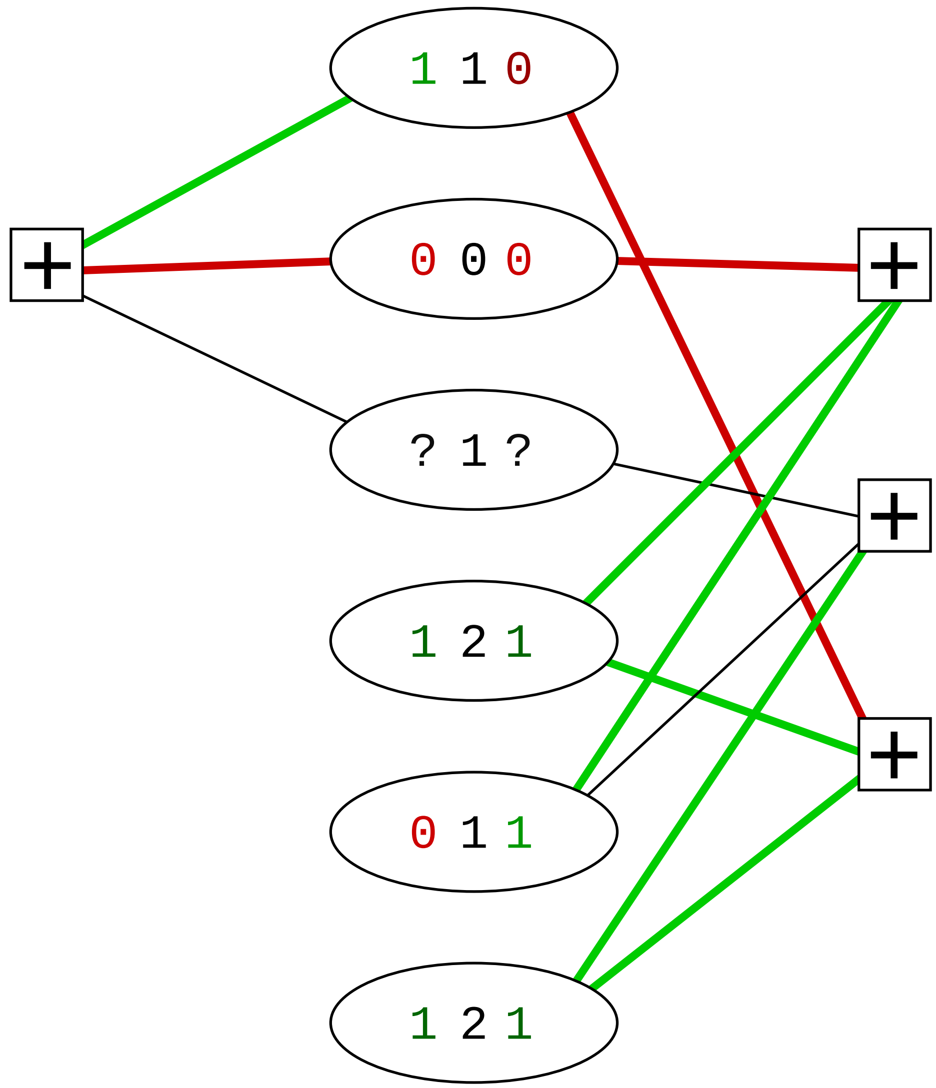
BAC Joint Decoding Example



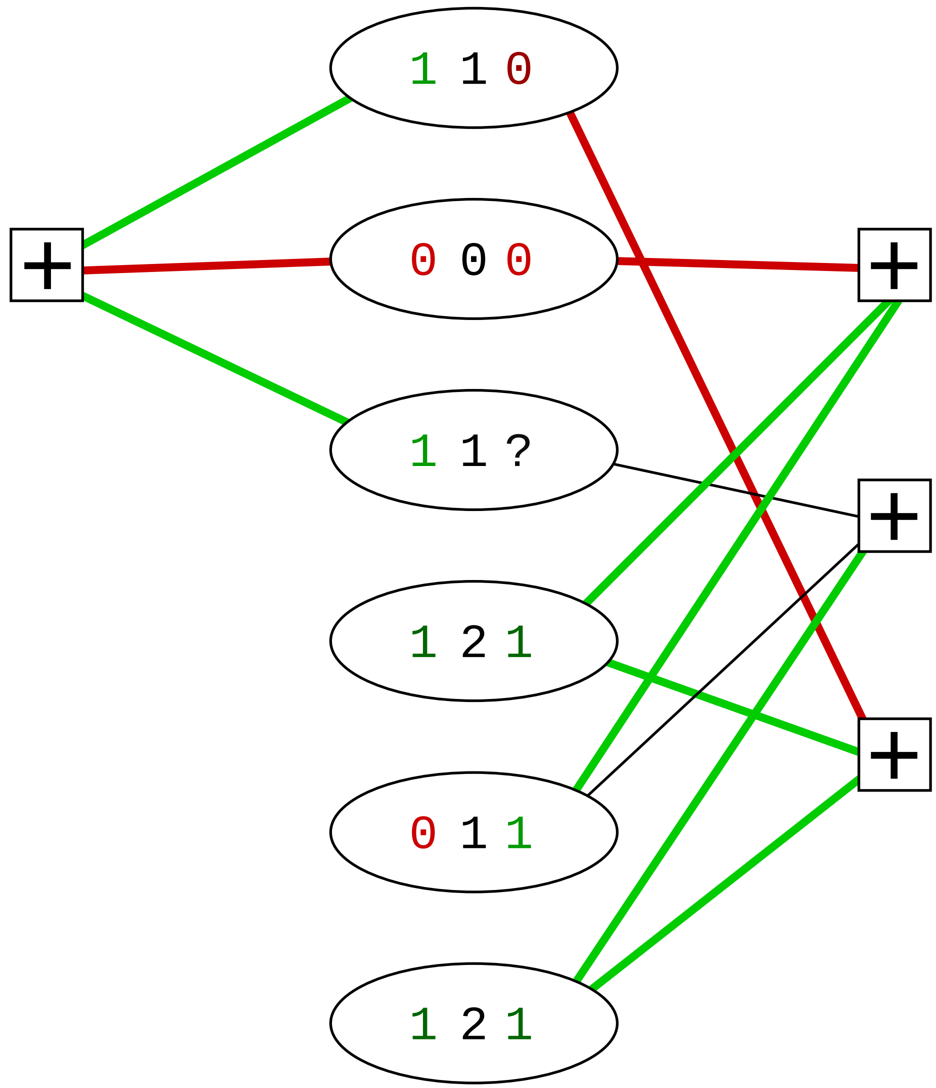
BAC Joint Decoding Example



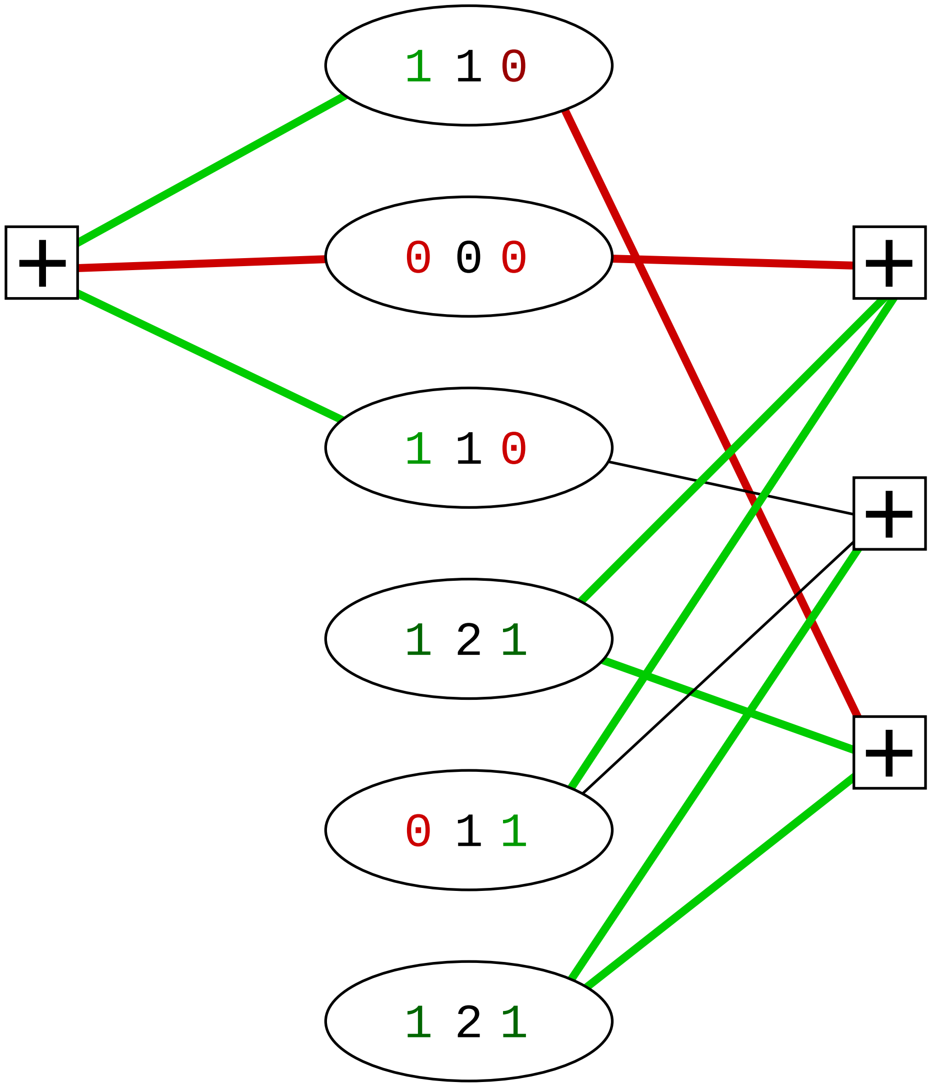
BAC Joint Decoding Example



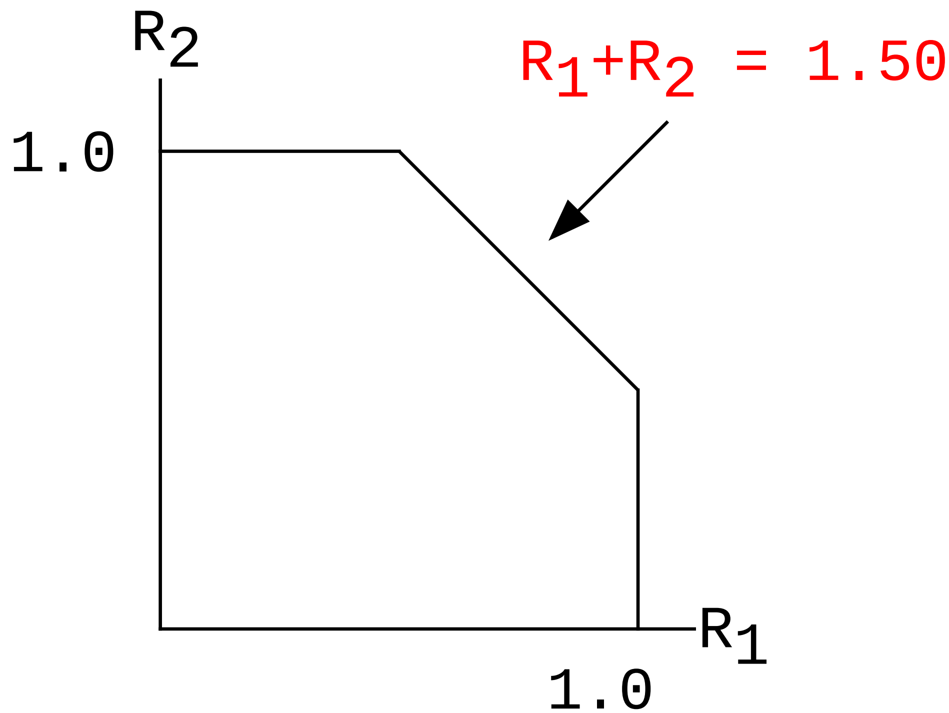
BAC Joint Decoding Example



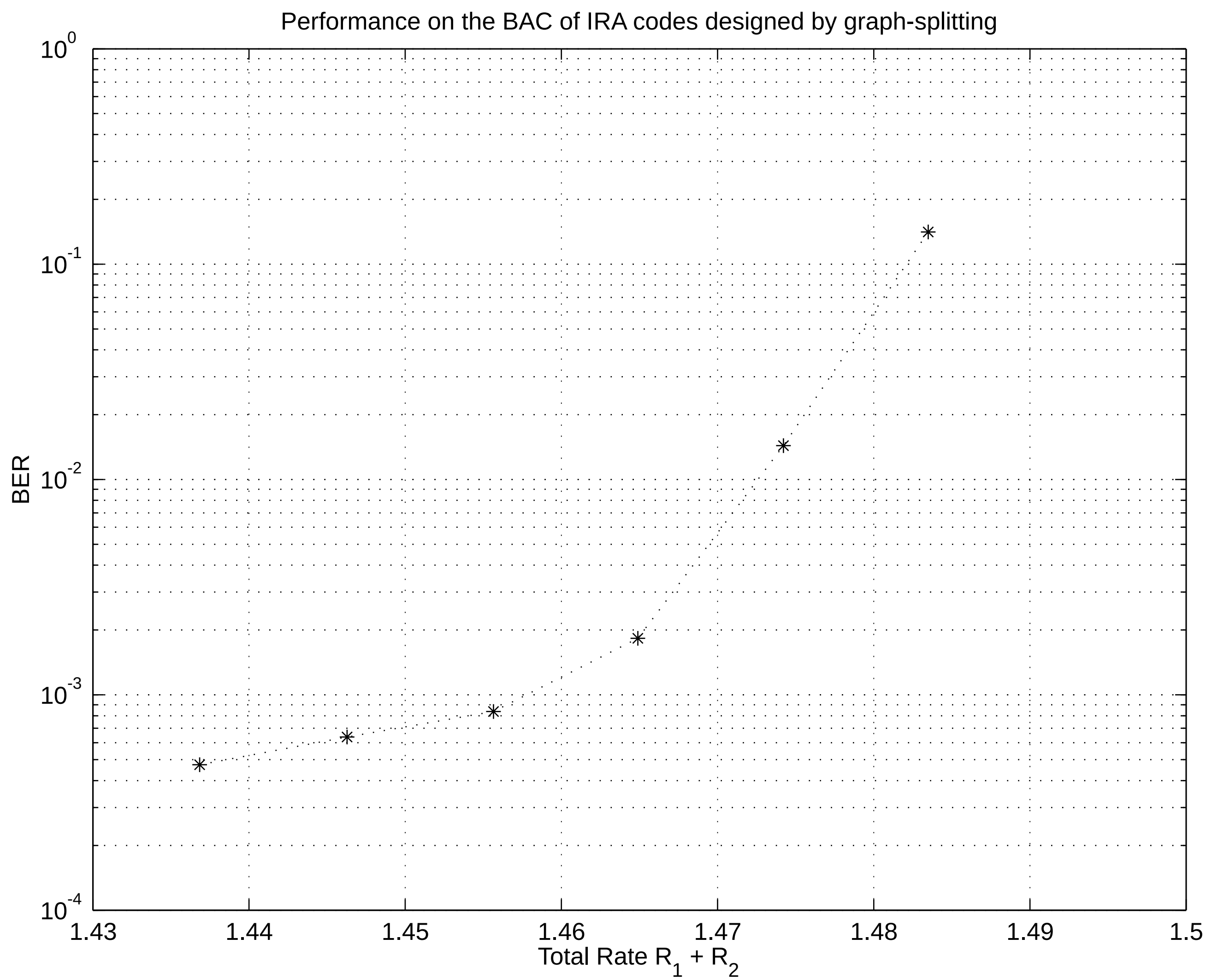
BAC Joint Decoding Example



The Capacity Region for the BAC



Experimental Results Based on Splitting Irregular RA Codes ($n = 10000$)



A Theorem.

Theorem. *Turbolike codes meet Shannon's Challenge on the BAC (without the need to timeshare).*

Proof. Use density evolution. It's almost exactly like the BEC.

(Palanki, Khandekar and McEliece, Allerton 2001)

Conclusions?

Conclusions?

- On *standard channel models* (SBIC's), coding technology is quite mature.

Conclusions?

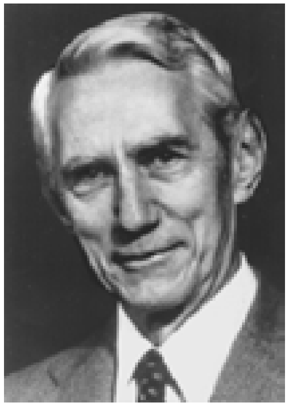
- On *standard channel models* (SBIC's), coding technology is quite mature.
- Still, *theory* has a lot of catching up to do. Density evolution may not be enough.

Conclusions?

- On *standard channel models* (SBIC's), coding technology is quite mature.
- Still, *theory* has a lot of catching up to do. Density evolution may not be enough.
- Although *applications* of turbolike codes to nonstandard channels are just beginning to appear, *graph-based iterative message-passing* may be a panacea.

Conclusions?

- On *standard channel models* (SBIC's), coding technology is quite mature.
- Still, *theory* has a lot of catching up to do. Density evolution may not be enough.
- Although *applications* of turbolike codes to nonstandard channels are just beginning to appear, *graph-based iterative message-passing* may be a panacea.



- Coding isn't dead quite yet!