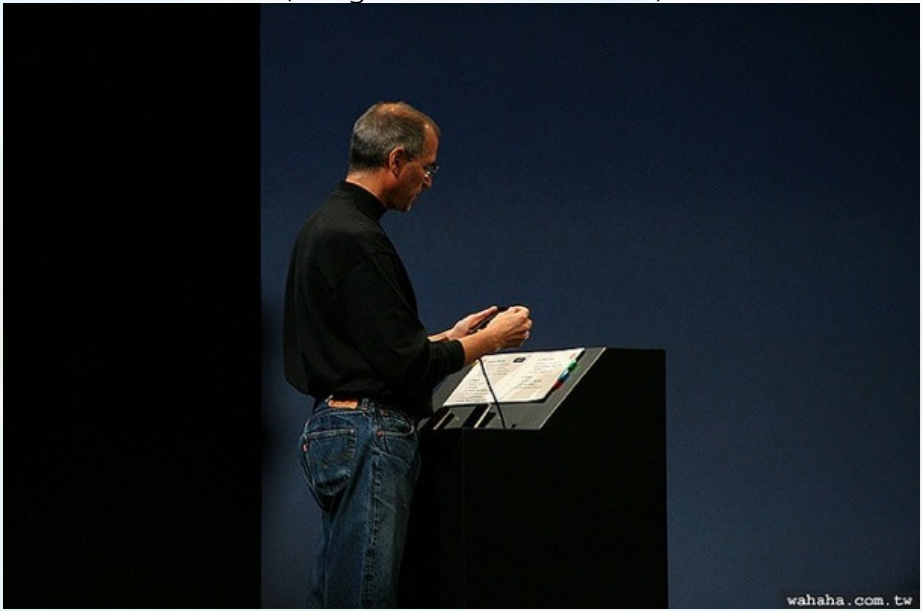
Channel Manipulation as a Coding Technique

Hsin-Po Wang (EECS, UC Berkeley)

What do the following pictures have in common?

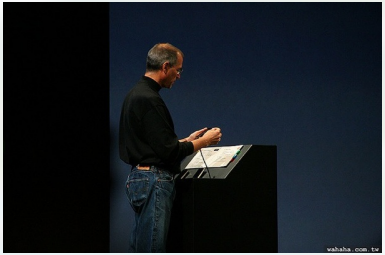
(Images from the internet)



(Images from the internet)

(Images from the internet)

(Images from the internet)



https://jobtalk.symbol.codes

https://jobta  mbol.codes

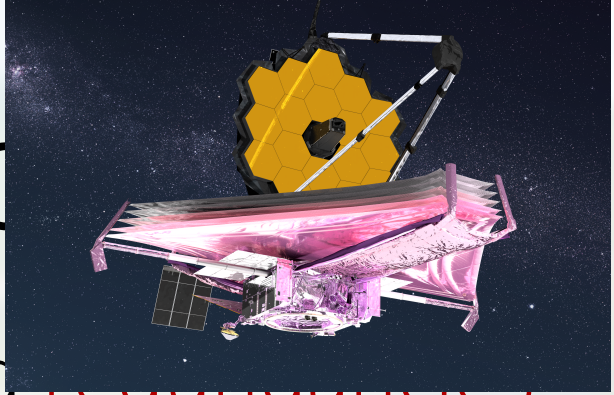
Channel Manipulation is Coding

H.-P Wang

Coding

= adding redundancies
in a smart way

Co



= adding redundancies
in a smart way

(Images from Wikipedia, NASA, IBM, Qualcomm)

= adding redundancies
in a smart way

(Images from Wikipedia, NASA, IBM, Qualcomm)

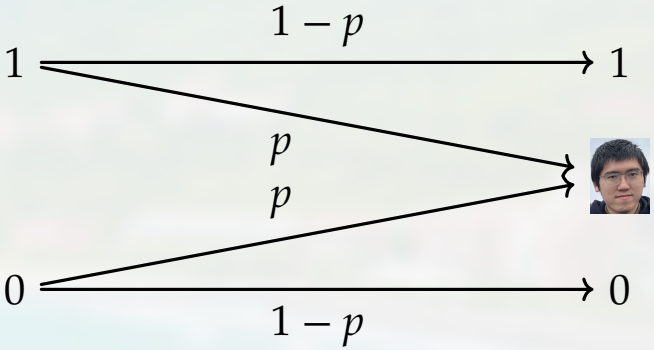
Redundancies
smart way

ipedia, NASA, IBM, Qualcomm)

Channel Manipulation is Coding

H.-P Wang

Math Framework of Coding



Binary erasure channel with erasure probability p: BEC(p)

N = 78 S = 83 Y = 89 U = 85

The “NSYSU” polynomial:

$$f(x) = 78 + 83x + 89x^2 + 83x^3 + 85x^4$$

N = 78 S = 83 Y = 89 U = 85

The “NSYSU” polynomial:

$$f(x) = 78 + 83x + 89x^2 + 83x^3 + 85x^4$$

$$f(-3) = 274 \quad f(-2) = 964 \quad f(-1) = 86 \quad f(0) = 78$$

$$f(1) = 418 \quad f(2) = 624 \quad f(4) = 254 \quad f(4) = 906$$

N = 78 S = 83 Y = 89 U = 85

The “NSYSU” polynomial:

$$f(x) = 7$$  $$.4$$

$f(-3) = 274$ $f(-2) = 964$ $f(-1) = 86$ $f(0) =$ 

$f(1) = 418$ $f(2) = 624$ $f(4) = 254$ $f(4) = 906$

N = 78 S = 83 Y = 89 U = 85

The “NSYSU” polynomial:

$$f(x) = 7$$  ^{4}

$f(-3) = 274$ $f(-2) = 964$ $f(-1) = 86$ $f(0) =$ 

$f(1) =$  $f(2) = 624$ $f(4) = 254$ $f(4) = 906$

N = 78 S = 83 Y = 89 U = 85

The “NSYSU” polynomial:

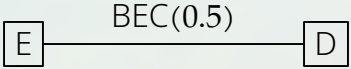
$f(x) = 7$  ^{4}

$f(-3) = 274$ $f(-2) = 964$ $f(-1) = 8$  $f(0) =$ 

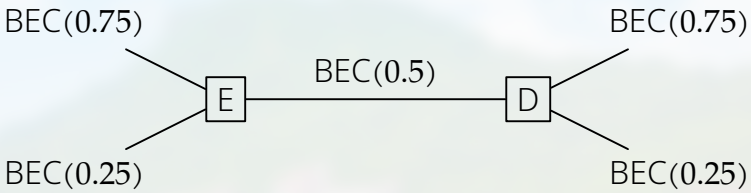
$f(1) = 4$  $f(2) = 624$ $f(4) = 254$ $f(4) = 906$

New Idea

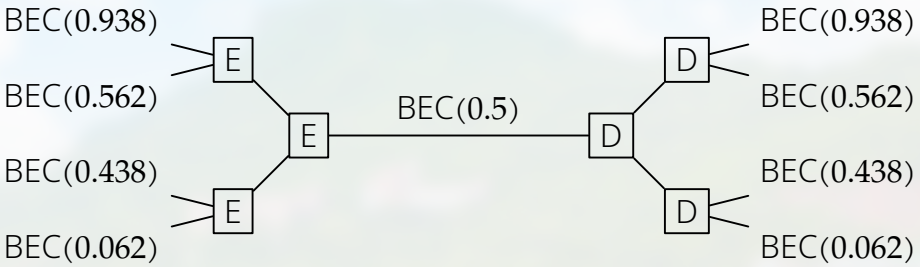
Polar Codes



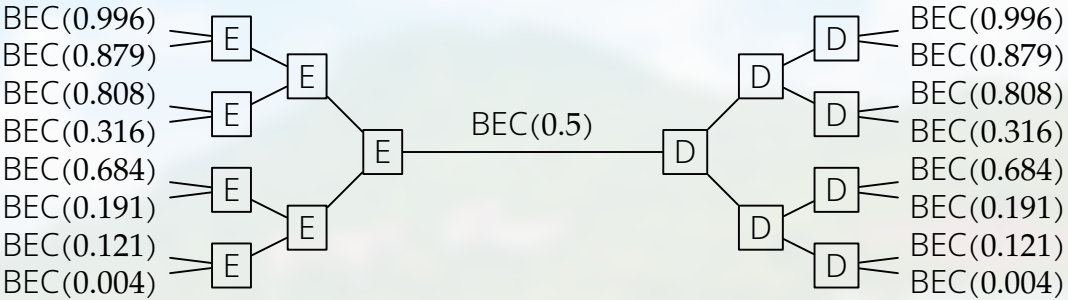
Suppose there are magic devices E and D that turns BEC(x) into BEC(x^2) and BEC($2x - x^2$).



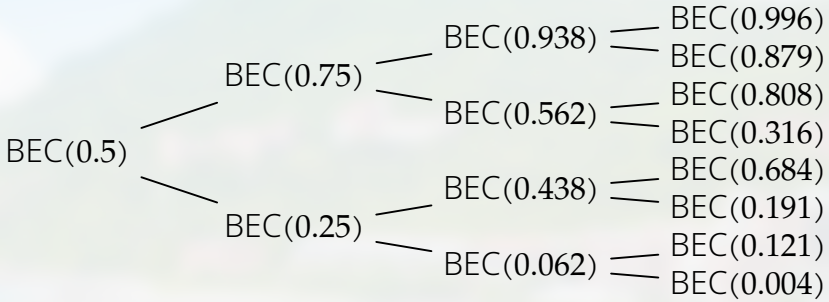
Suppose there are magic devices E and D that turns BEC(x) into BEC(x^2) and BEC($2x - x^2$).



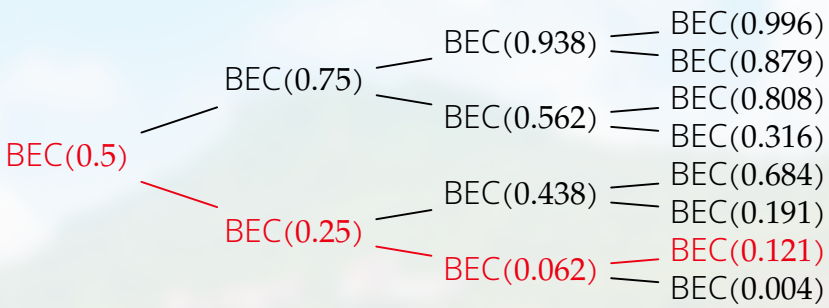
What if we apply apply more magic devices?



What if we apply apply more magic devices?
And more and more and more???

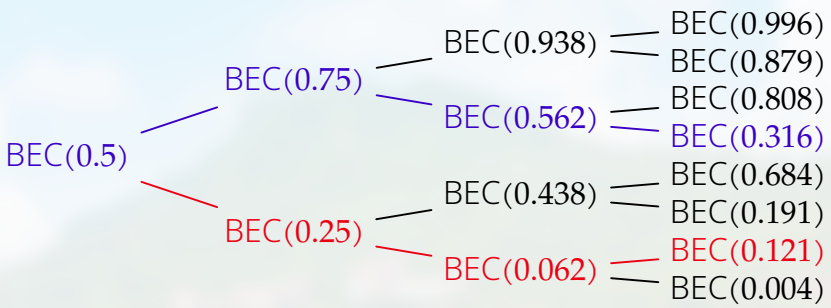


This is a tree



: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$



: This is a martingale

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$

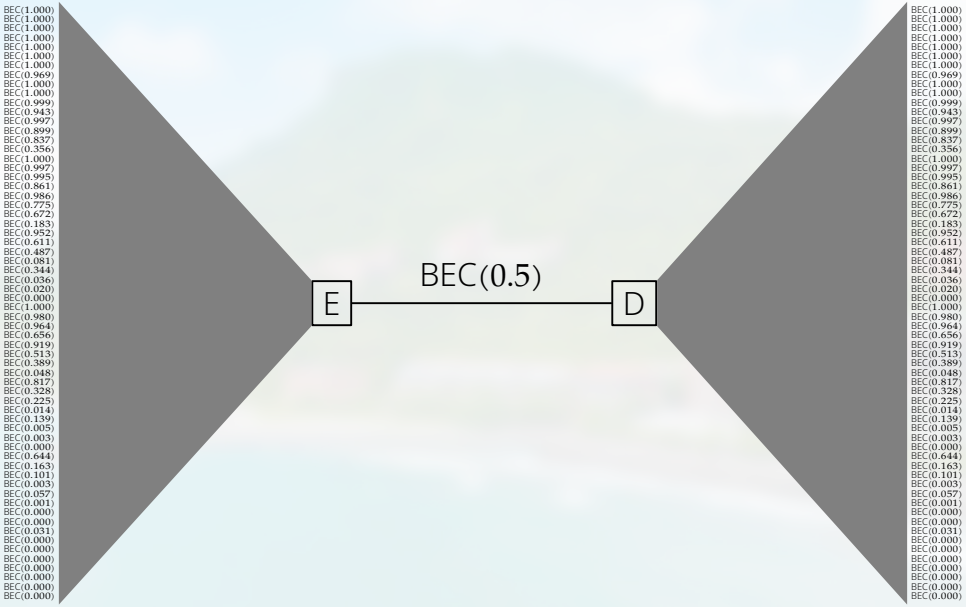
: Bounded martingale converges.

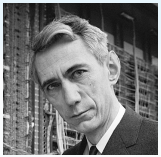
$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$

: Bounded martingale converges.

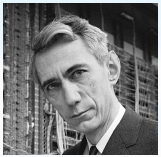
: BECs converges to 0 or 1.

$$M_{n+1} := \begin{cases} 2M_n - M_n^2 & \text{with prob. } 1/2, \\ M_n^2 & \text{with prob. } 1/2. \end{cases}$$


BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(0.969)
BEC(1.000)
BEC(1.000)
BEC(0.999)
BEC(0.943)
BEC(0.997)
BEC(0.899)
BEC(0.837)
BEC(0.356)
BEC(1.000)
BEC(0.997)
BEC(0.995)
BEC(0.861)
BEC(0.986)
BEC(0.775)
BEC(0.672)
BEC(0.183)
BEC(0.952)
BEC(0.611)
BEC(0.487)
BEC(0.081)
BEC(0.344)
BEC(0.036)
BEC(0.020)
BEC(0.000)
BEC(1.000)
BEC(0.980)
BEC(0.964)
BEC(0.656)
BEC(0.919)
BEC(0.513)
BEC(0.389)
BEC(0.048)
BEC(0.817)
BEC(0.328)
BEC(0.225)
BEC(0.014)
BEC(0.139)
BEC(0.005)
BEC(0.003)
BEC(0.000)
BEC(0.644)
BEC(0.163)
BEC(0.101)
BEC(0.003)
BEC(0.057)
BEC(0.001)
BEC(0.000)
BEC(0.000)
BEC(0.031)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
E
BEC(0.5)
D
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(1.000)
BEC(0.969)
BEC(1.000)
BEC(1.000)
BEC(0.999)
BEC(0.943)
BEC(0.997)
BEC(0.899)
BEC(0.837)
BEC(0.356)
BEC(1.000)
BEC(0.997)
BEC(0.995)
BEC(0.861)
BEC(0.986)
BEC(0.775)
BEC(0.672)
BEC(0.183)
BEC(0.952)
BEC(0.611)
BEC(0.487)
BEC(0.081)
BEC(0.344)
BEC(0.036)
BEC(0.020)
BEC(0.000)
BEC(1.000)
BEC(0.980)
BEC(0.964)
BEC(0.656)
BEC(0.919)
BEC(0.513)
BEC(0.389)
BEC(0.048)
BEC(0.817)
BEC(0.328)
BEC(0.225)
BEC(0.014)
BEC(0.139)
BEC(0.005)
BEC(0.003)
BEC(0.000)
BEC(0.644)
BEC(0.163)
BEC(0.101)
BEC(0.003)
BEC(0.057)
BEC(0.001)
BEC(0.000)
BEC(0.000)
BEC(0.031)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)
BEC(0.000)

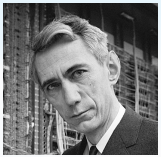


: 5000 bits / 10000 channel uses



: 5000 bits / 10000 channel uses

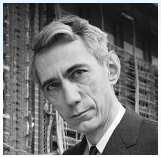
: Polar code 490bits/10000uses



: 5000 bits / 10000 channel uses

: Polar code 490bits/10000uses

In general, $(1/2 - \varepsilon)N$ bits / N uses



: 5000 bits / 10000 channel uses

: Polar code 490bits/10000uses

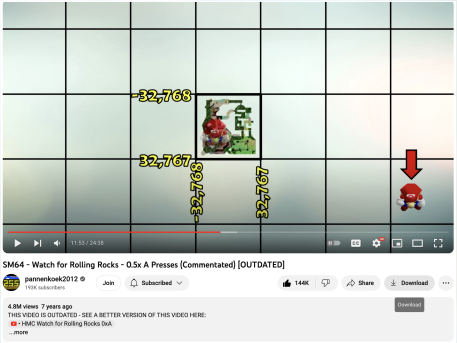
In general, $(1/2 - \varepsilon)N$ bits / N uses

THE END?

5G is for

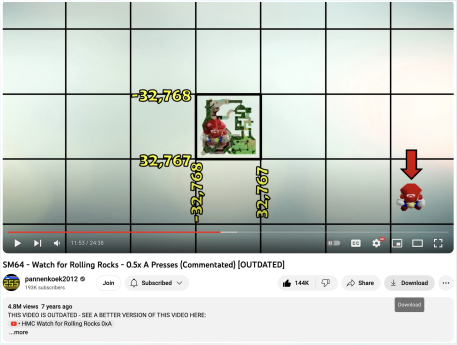
H.-P. Wang

5G is for



video streaming

5G is for

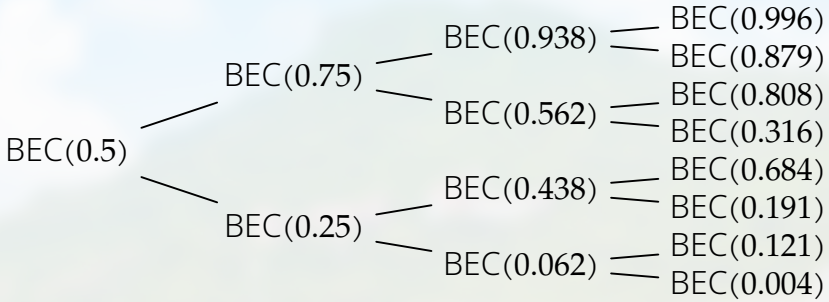


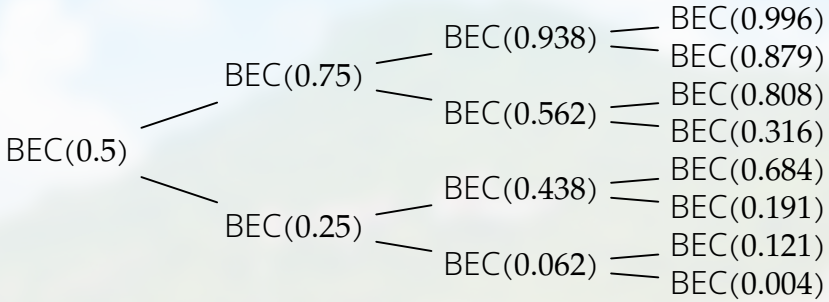
video streaming

flying drone

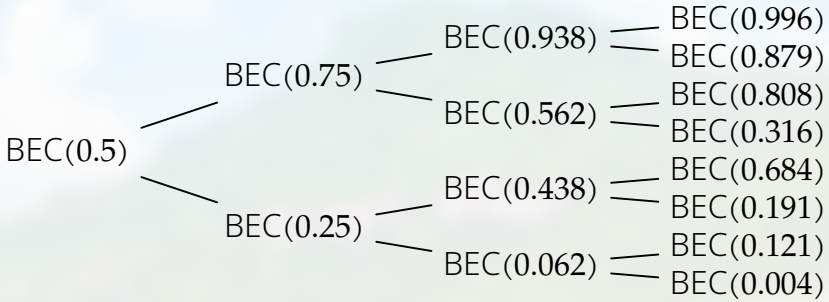
Channel Manipulation is Coding

H.-P. Wang


BEC(0.5)
BEC(0.75)
BEC(0.25)
BEC(0.938)
BEC(0.562)
BEC(0.438)
BEC(0.062)
BEC(0.996)
BEC(0.879)
BEC(0.808)
BEC(0.316)
BEC(0.684)
BEC(0.191)
BEC(0.121)
BEC(0.004)

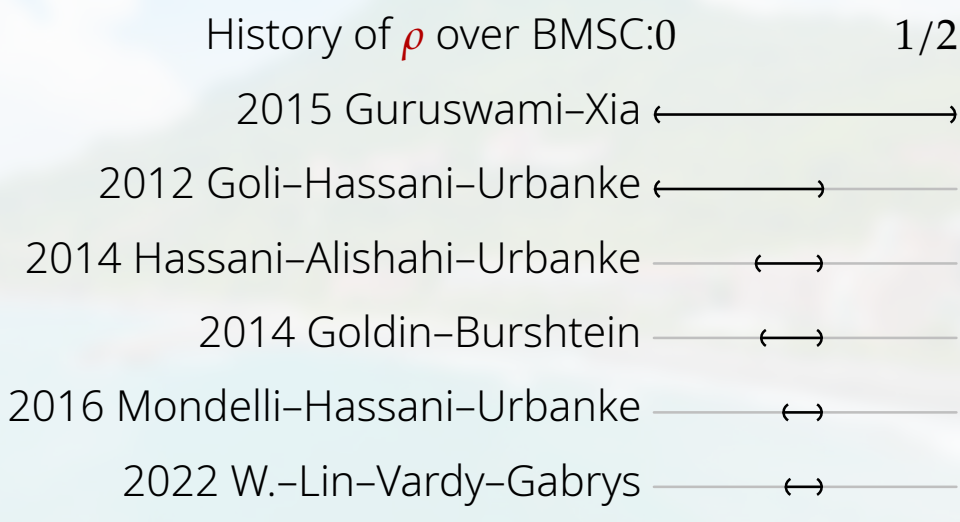


Early channels are not useful.



Early channels are not useful. How to accelerate polarization?

$(1/2 - N^{-\rho})N$ bits / N channel uses




Improve ρ over BEC: 0
1/2
2010 Hassani–Alishahi–Urbanke 2×2
2010 Korada–Montanari–Telatar–Urbanke 2×2
2014 Fazeli–Vardy 8×8
2021 Trofimiuk–Trifonov 16×16
2022 Duursma–Gabrys–Guruswami–Lin–W. 2×2/GF4
2021 Trofimiuk 24×24
2021 Yao–Fazeli–Vardy 32×32
2021 Yao–Fazeli–Vardy 64×64

The optimal ρ: 0 $1/2$

2019 Pfister–Urbanke
q-ary erasure channel, $q \to \infty$

2021 Fazeli–Hassani–Mondelli–Vardy
binary erasure channel

2022 Guruswami–Riazanov–Ye
binary symmetric memoryless channel

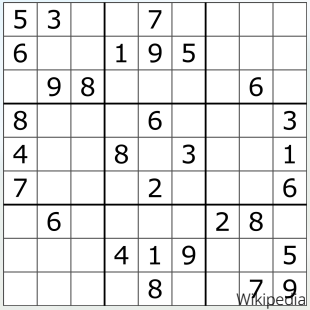
2021 W.–Duursma
discrete memoryless channel

2011 Alamdar-Yazdi–Kschischang:
Orune the tree to reduce complexity.

2017 El-Khamy–Mahdavifar–Feygin–Lee–Kang:
Oruning reduces complexity by a scalar;
still $O(N \log N)$.

2021 W.–Duursma: $O(N \log \log N)$
if relax the performance requirement.

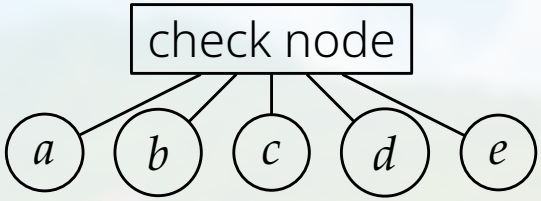
Trade-off: complexity $\approx O(N \log(-\log(\text{decode error})))$.

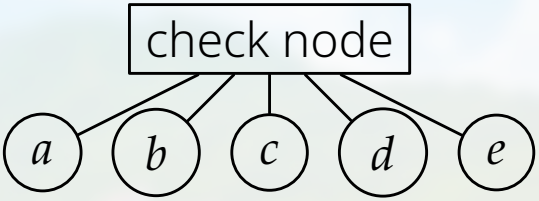


Wikipedia

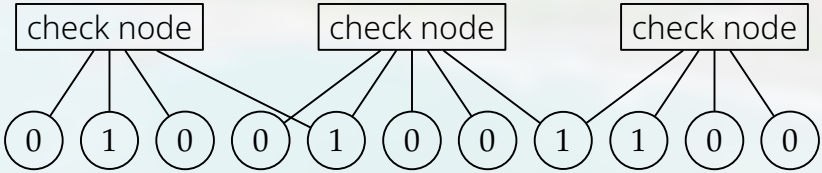
Low-Density Parity-Check (LDPC) Codes

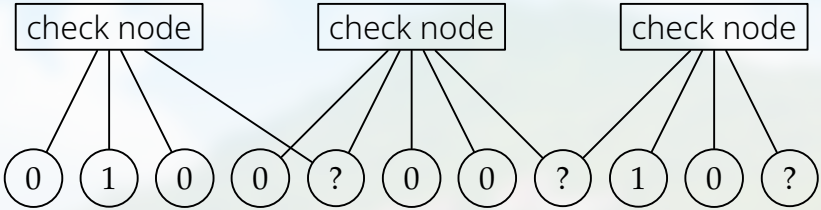
Rule: Every Sum to an even number

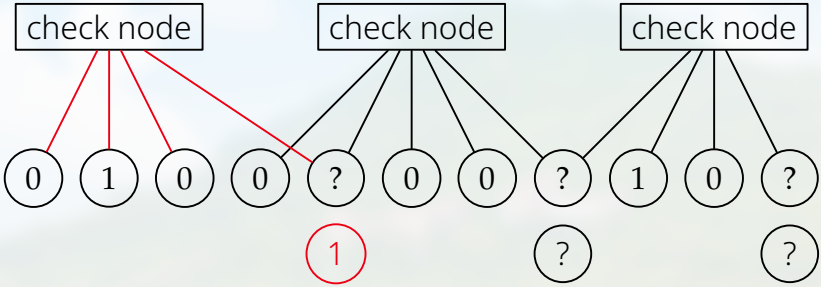


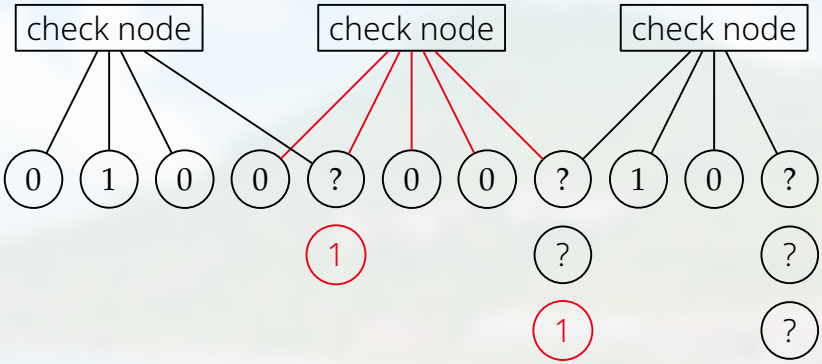


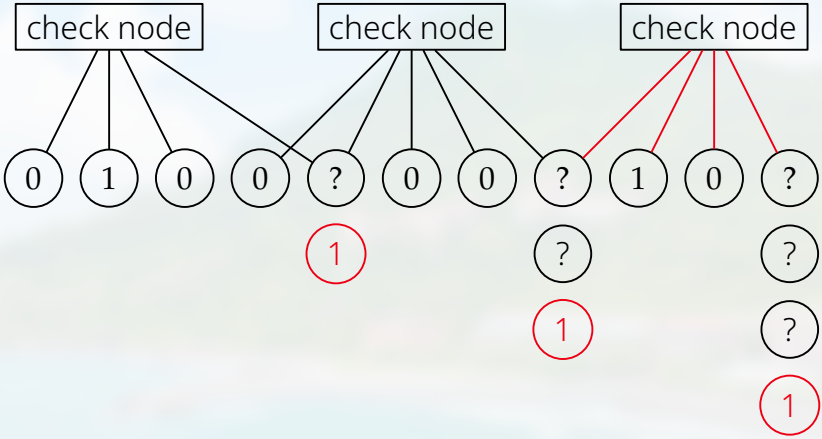
Rule: Every Sum to an even number

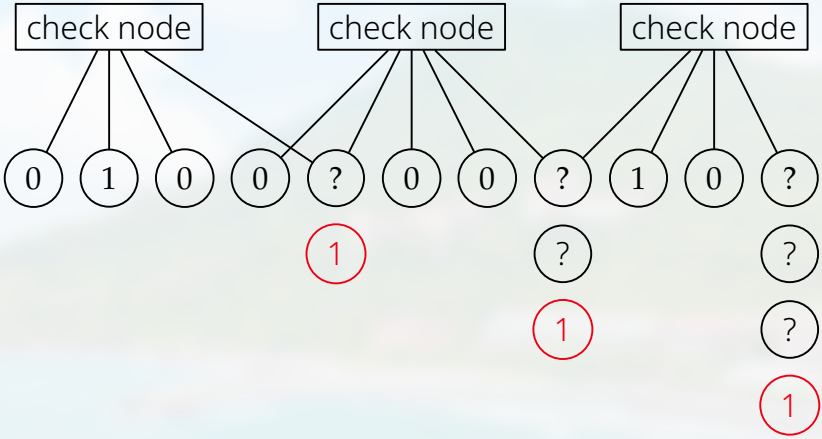



check node
check node
check node
0
1
0
0
?
0
0
?
1
0
?

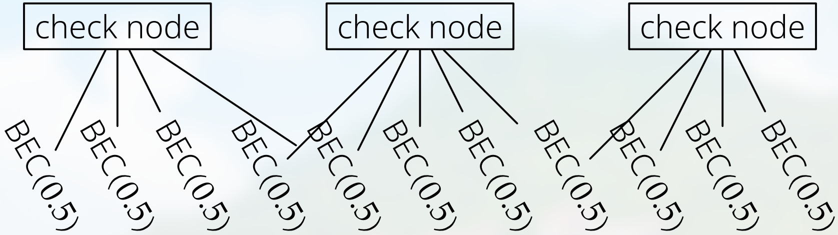

check node
check node
check node
0
1
0
0
?
0
0
?
1
0
?
1
?
?

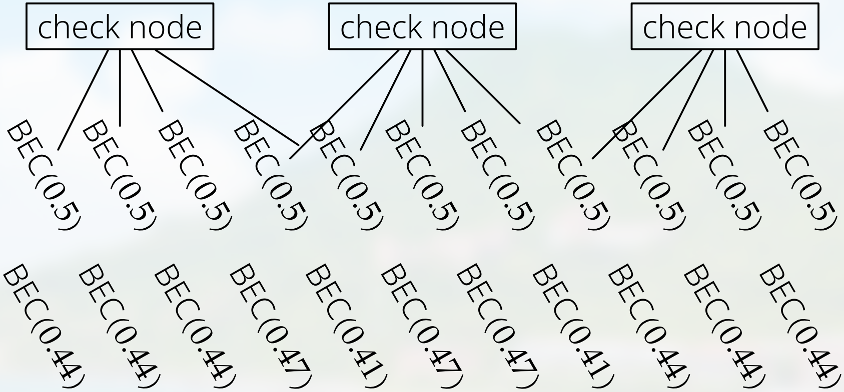

check node
check node
check node
0
1
0
0
?
0
0
?
1
0
?
1
?
?
1
?
?

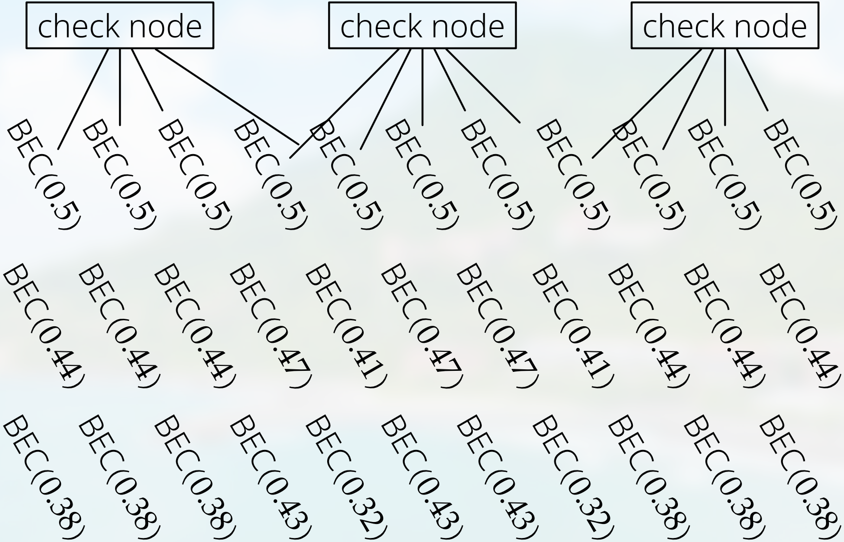

check node
check node
check node
0
1
0
0
?
0
0
?
1
0
?
1
?
1
?
?
1



How to analyze, mathematically?


check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)


check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.47)
BEC(0.41)
BEC(0.47)
BEC(0.47)
BEC(0.41)
BEC(0.44)
BEC(0.44)
BEC(0.44)


check node
check node
check node
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.5)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.47)
BEC(0.41)
BEC(0.47)
BEC(0.47)
BEC(0.41)
BEC(0.44)
BEC(0.44)
BEC(0.44)
BEC(0.38)
BEC(0.38)
BEC(0.38)
BEC(0.43)
BEC(0.32)
BEC(0.43)
BEC(0.43)
BEC(0.32)
BEC(0.38)
BEC(0.38)
BEC(0.38)


0:05 / 0:39
How To Make An Ice Bubble
Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Daniells
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

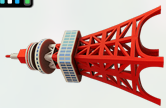
How To Make An Ice Bubble

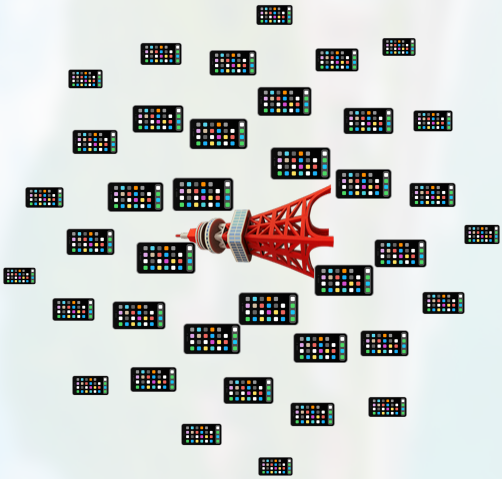
Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

How To Make An Ice Bubble

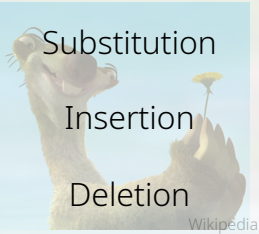
Rosemary Danielis
690 subscribers
Subscribe
732
Share
Download

Challenges





DNA coding



Permutation