Excerpt from

An Introduction to Low-Density Parity-Check Codes

Paul H. Siegel

Electrical and Computer Engineering

University of California, San Diego

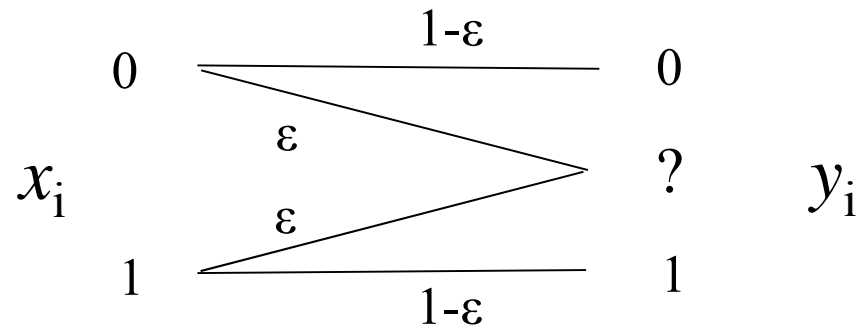Center for Wireless Communications

CMRR
Center for Magnetic Recording Research

Decoding for the BEC

- Recall: Binary erasure channel, BEC(ε)

$$x = (x_1, x_2, \ldots, x_n) \qquad \text{transmitted codeword}$$
$$y = (y_1, y_2, \ldots, y_n) \qquad \text{received word}$$

- **Note:** if $y_i \in \{0,1\}$, then $x_i = y_i$.

Local Decoding of Erasures

- $d_{min} = 3$, so any two erasures can be uniquely filled to get a codeword.

- Decoding can be done *locally*: Given any pattern of one or two erasures, there will always be a parity-check (circle) involving exactly one erasure.

- The parity-check represented by the circle can be used to fill in the erased bit.

- This leaves at most one more erasure. Any parity-check (circle) involving it can be used to fill it in.

LDPC Codes
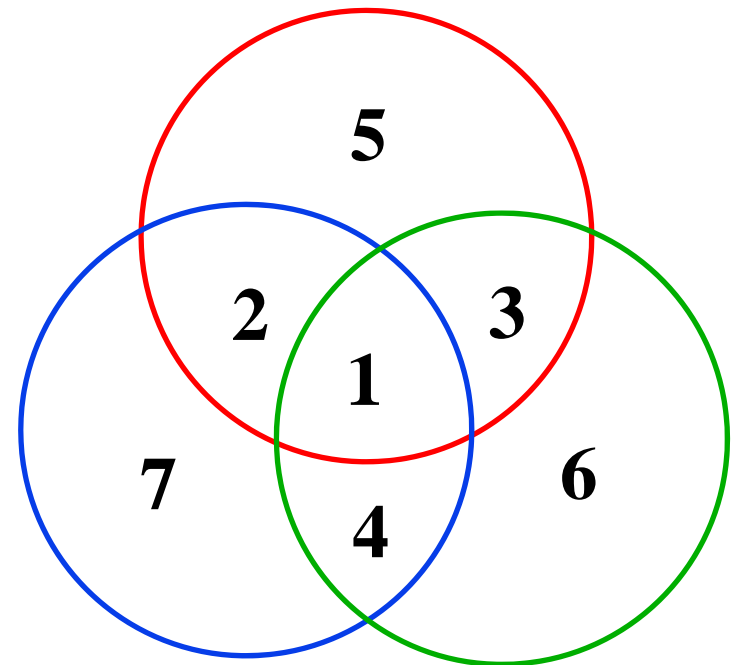
Local Decoding - Example

- All-0's codeword transmitted.

- Two erasures as shown.

- Start with either the red parity or green parity circle.

- The red parity circle requires that the erased symbol inside it be 0.

LDPC Codes
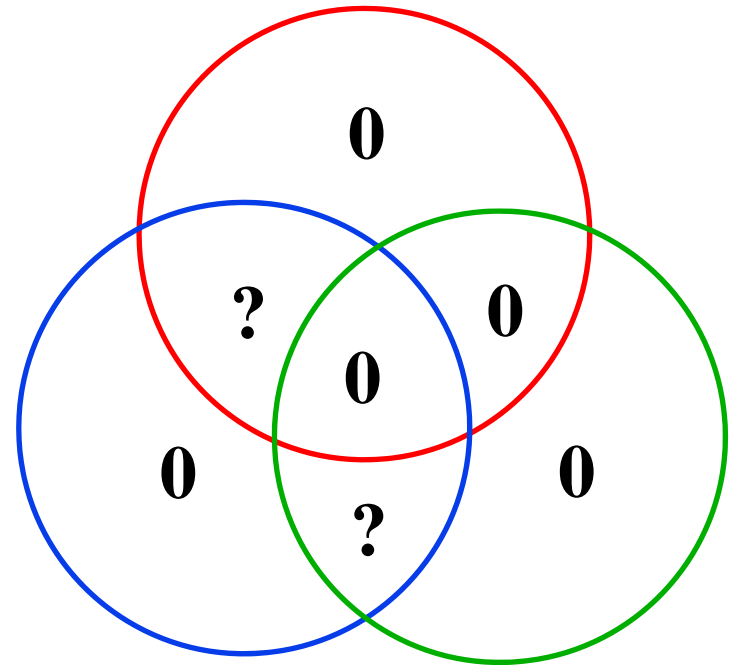
Local Decoding -Example

- Next, the green parity circle or the blue parity circle can be selected.

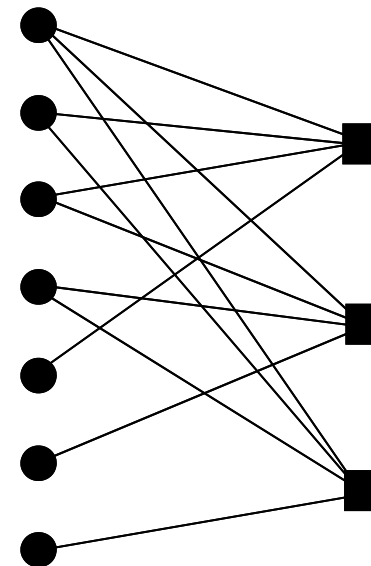- Either one requires that the remaining erased symbol be 0.

LDPC Codes
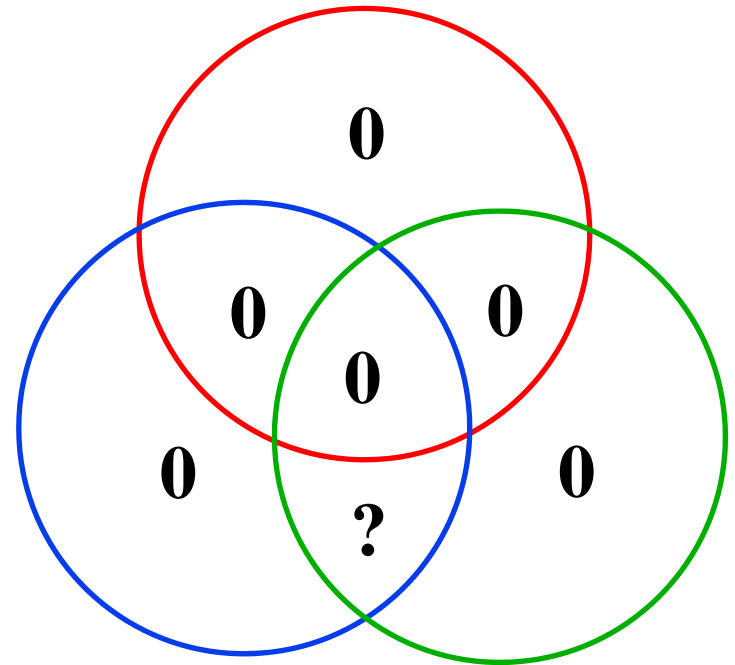
Local Decoding -Example

- Estimated codeword:

  [0 0 0 0 0 0 0]

- Decoding successful!!

- This procedure would have worked no matter which codeword was transmitted.
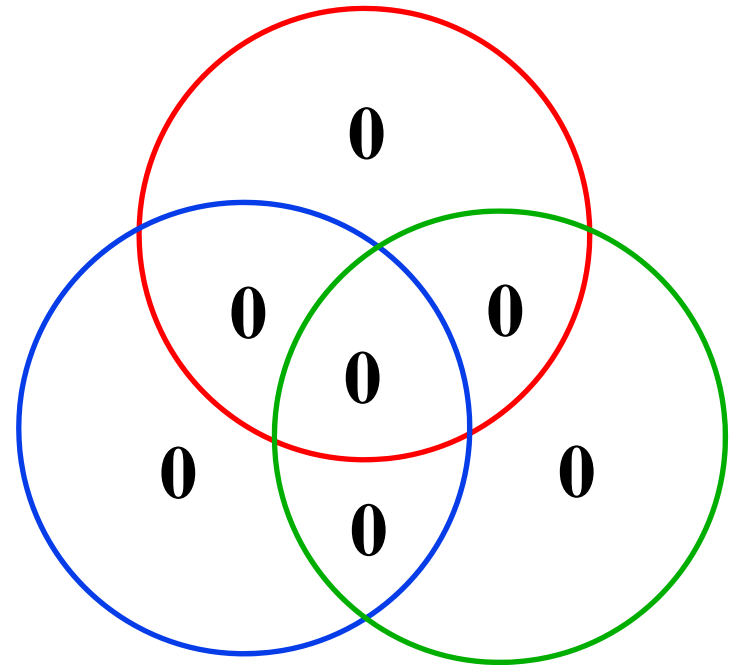
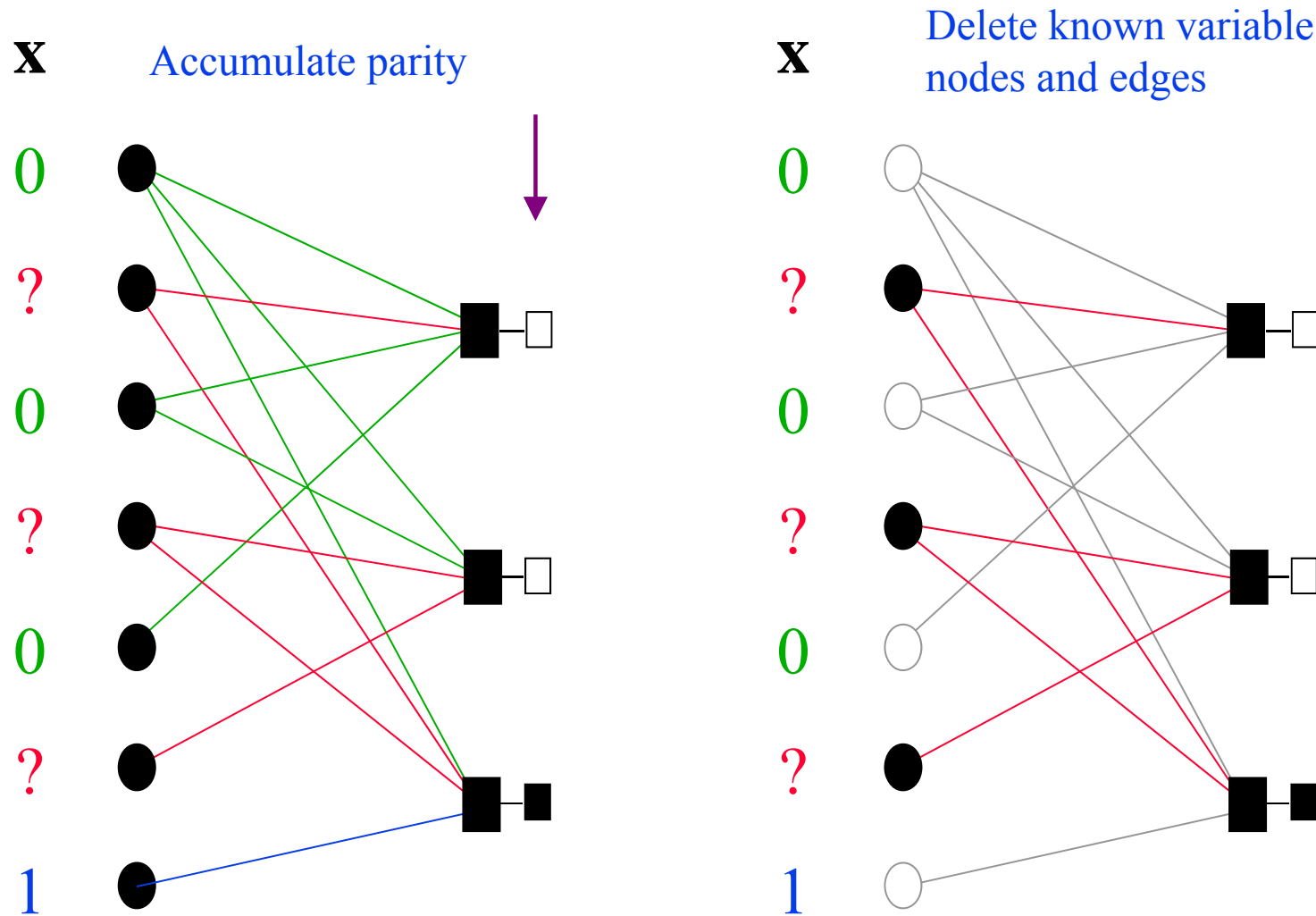# *Decoding with the Tanner Graph:*
an a-Peeling Decoder

- ## Initialization:

  - Forward known variable node values along outgoing edges

  - Accumulate forwarded values at check nodes and "record" the parity

  - Delete known variable nodes and all outgoing edges

LDPC Codes

Peeling Decoder – Initialization

LDPC Codes

Peeling Decoder – Initialization

x Accumulate parity

0
?
0
?
0
?
1

x Delete known variable nodes and edges
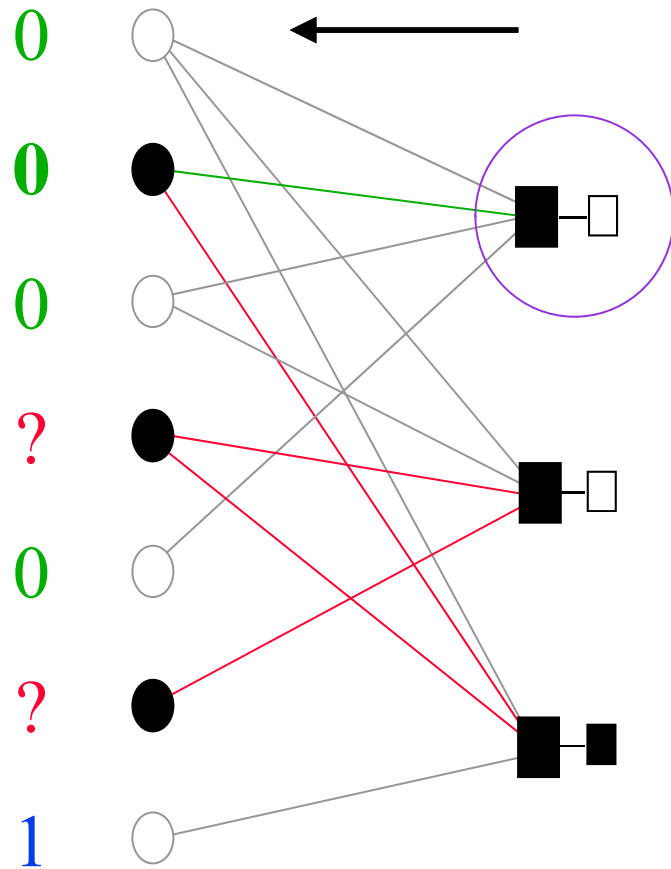
0
?
0
?
0
?
1

LDPC Codes
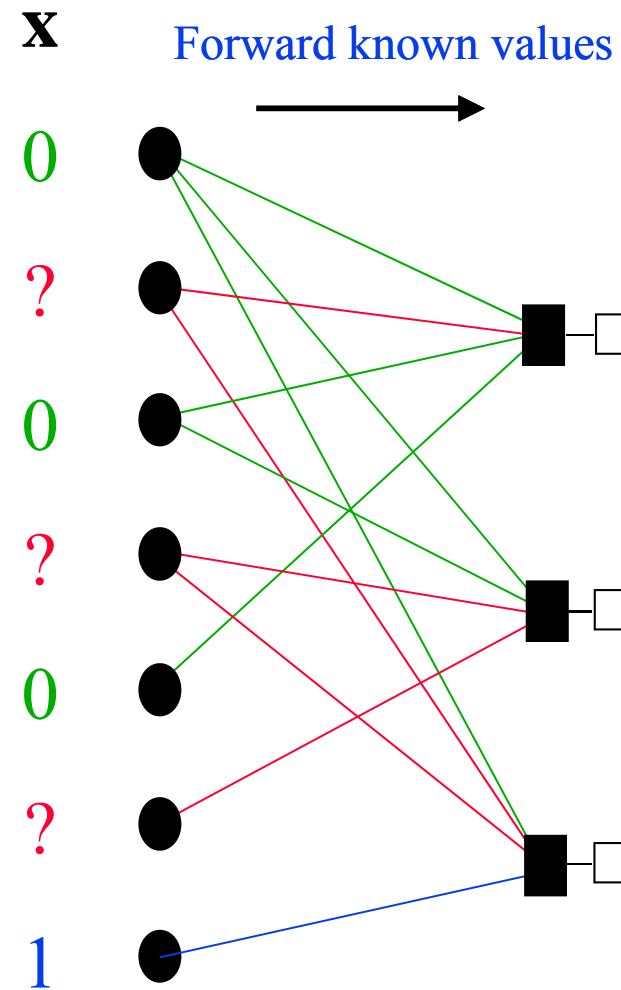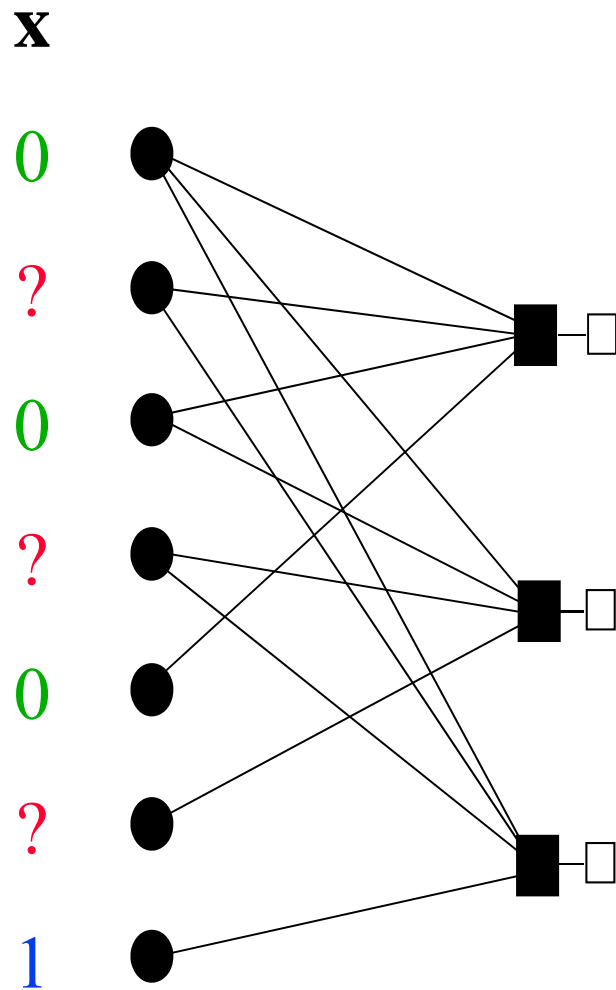
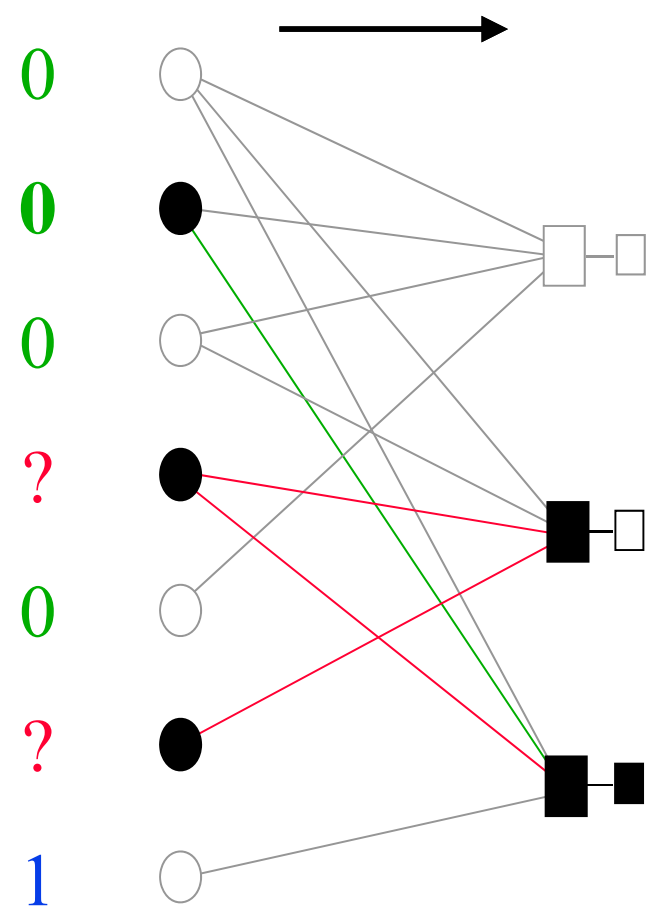# *Decoding with the Tanner Graph:*
an a-Peeling Decoder

- ## Decoding step:
  - Select, if possible, a check node with one edge remaining; forward its parity, thereby determining the connected variable node
  - Delete the check node and its outgoing edge
  - Follow procedure in the initialization process at the known variable node

- ## Termination
  - If remaining graph is empty, the codeword is determined
  - If decoding step gets stuck, declare decoding failure
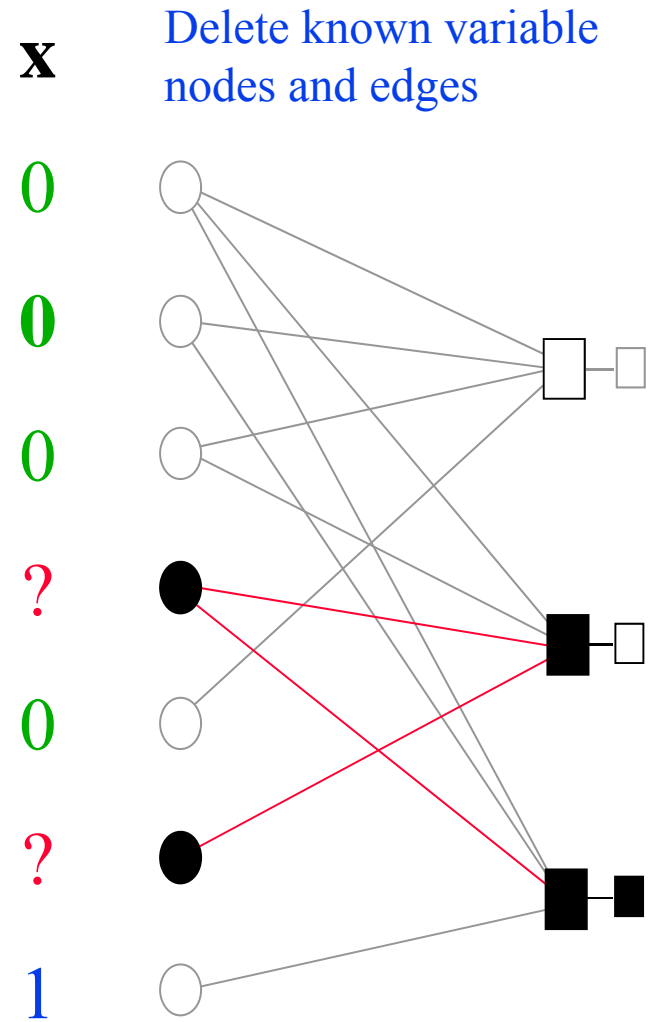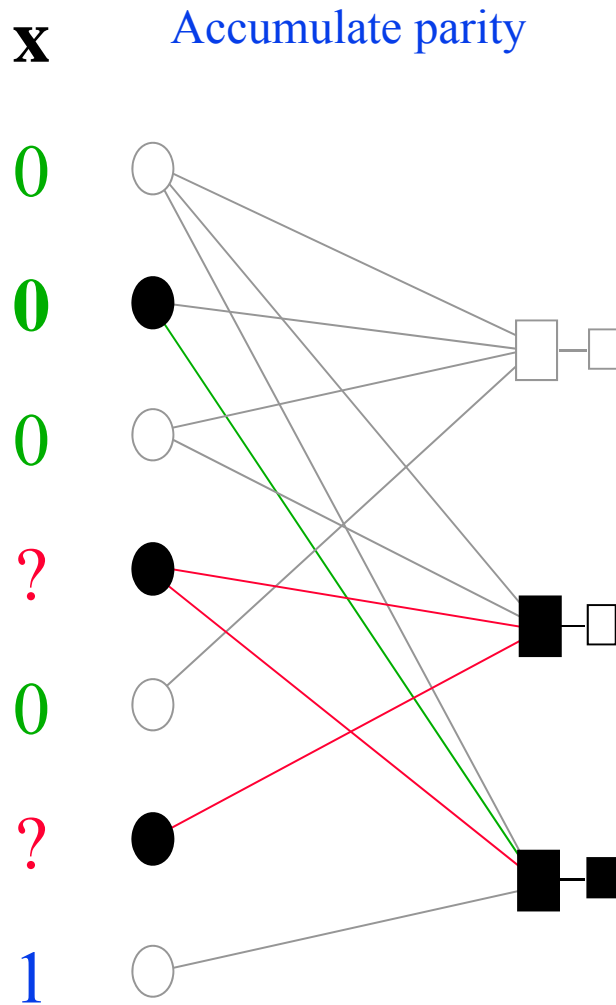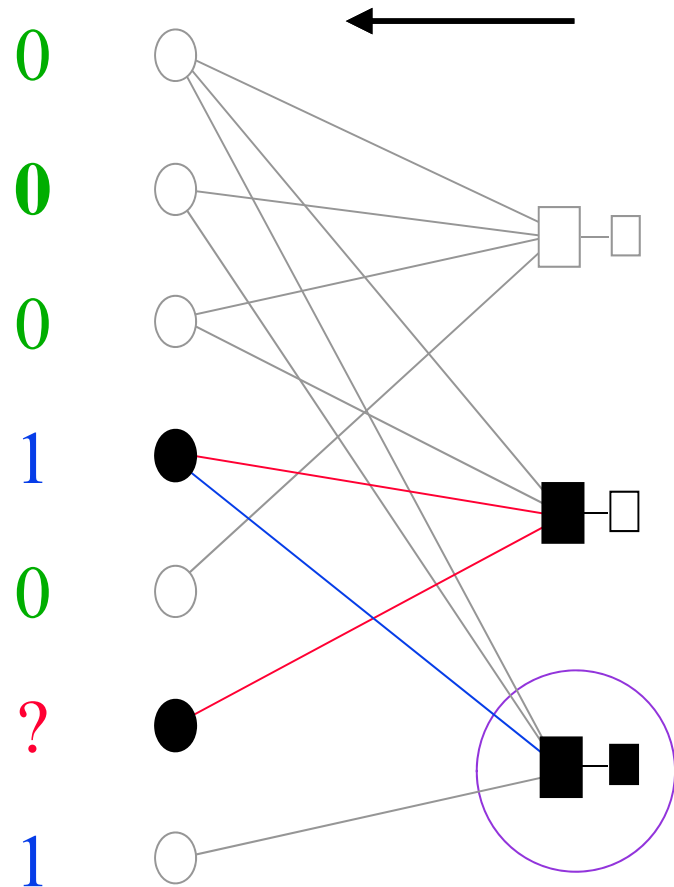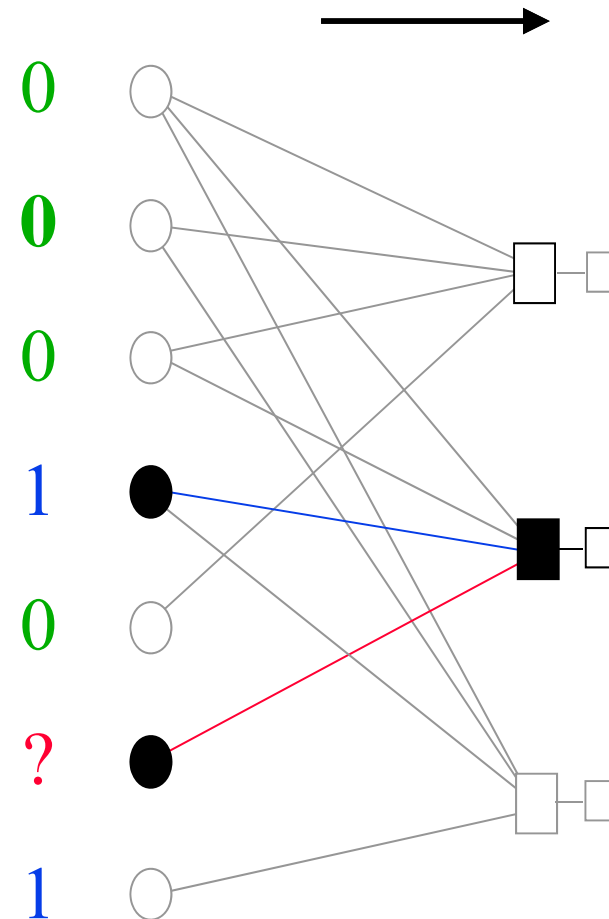
Peeling Decoder – Step 1

x — Find degree-1 check node; forward accumulated parity; determine variable node value

0
0
0
?
0
?
1

x — Delete check node and edge; forward new variable node value

0
0
0
?
0
?
1

LDPC Codes

Peeling Decoder – Step 1

x Accumulate parity

x Delete known variable nodes and edges

0

0

0

?

0

?

1

LDPC Codes

Peeling Decoder – Step 2

LDPC Codes

Peeling Decoder – Step 2

Accumulate parity

Delete known variable
nodes and edges

LDPC Codes

Peeling Decoder – Step 3

x

Find degree-1 check node;
forward accumulated parity;
determine variable node value

0

0

0

1

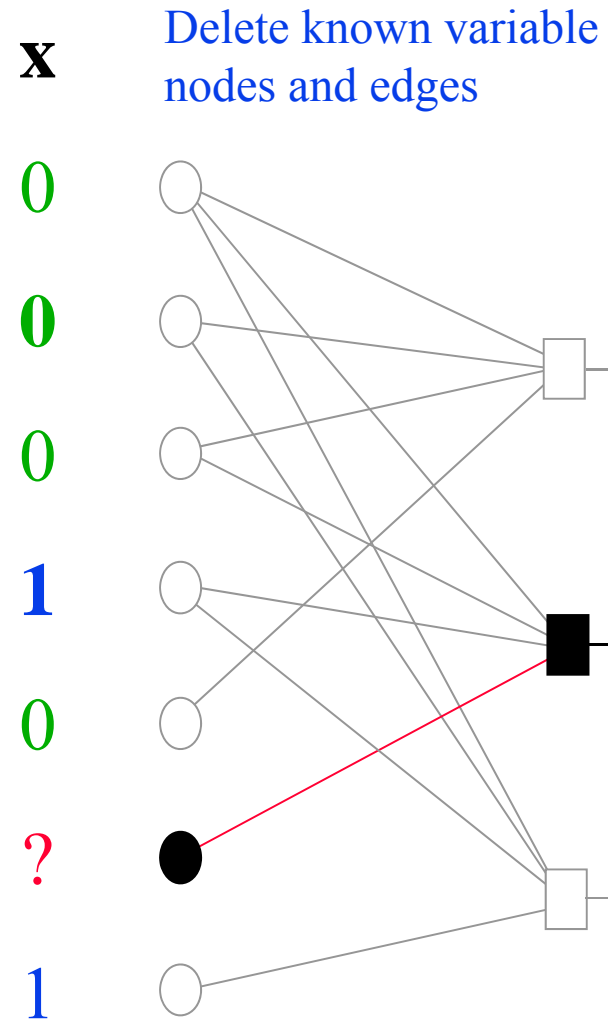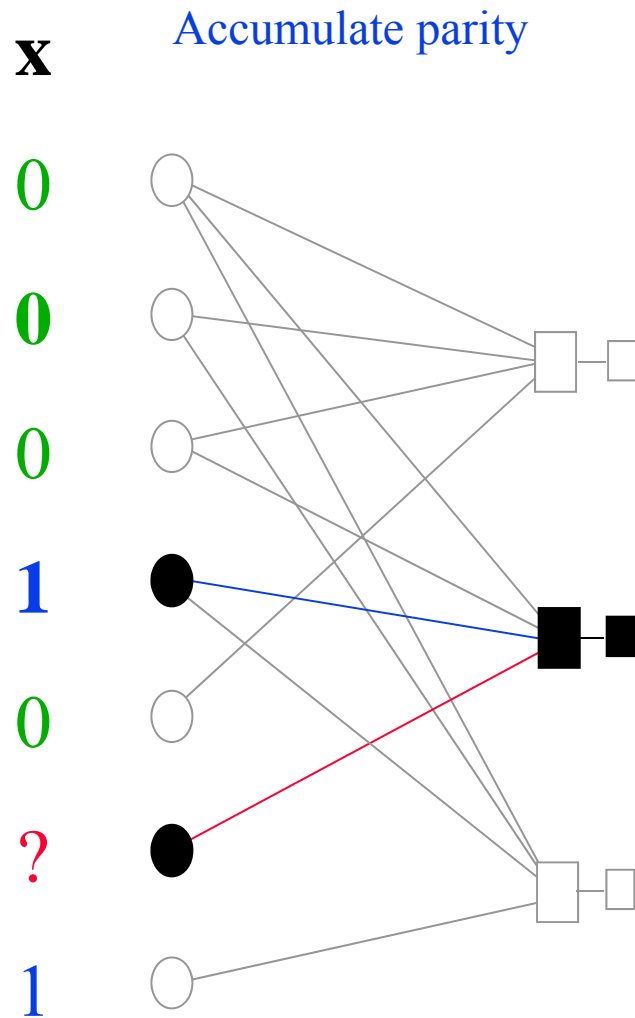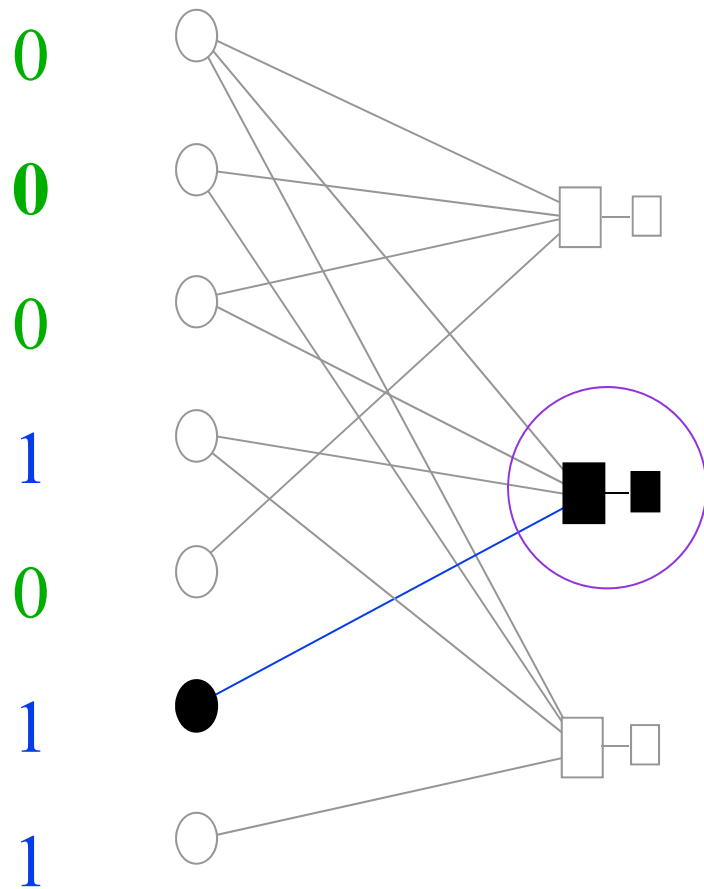0

1

1

x

Delete check node and edge;
decoding complete

→

0

0

0

1

0

1

1

LDPC Codes
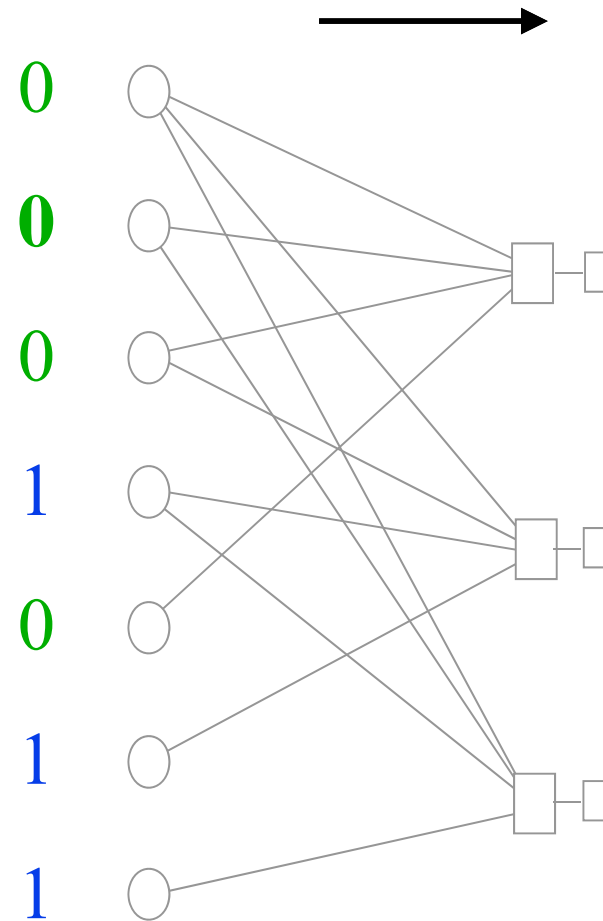
Message-Passing Decoding

- The local decoding procedure can be described in terms of an iterative, "message-passing" algorithm in which all variable nodes and all check nodes in parallel iteratively pass messages along their adjacent edges.

- The values of the code bits are updated accordingly.

- The algorithm continues until all erasures are filled in, or until the completion of a specified number of iterations.

LDPC Codes