Capacity-Achieving Codes for the BEC with Bounded Complexity

Henry D. Pfister

In collaboration with Igal Sason and Rüdiger Urbanke

Texas A&M University
April 17th, 2006

Outline

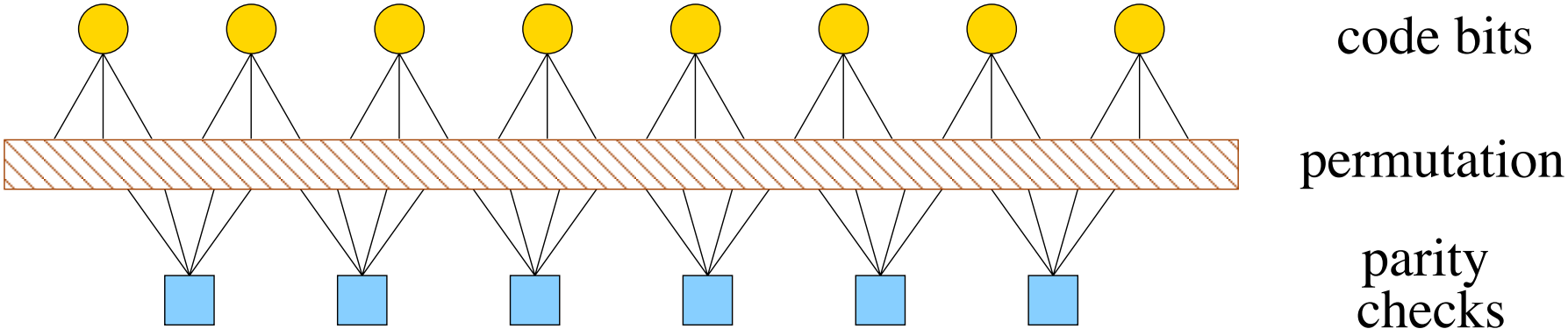
Introduction

- Binary Erasure Channel (BEC)
  - Each bit sent perfectly (with prob. $1-p$) or erased (with prob. p)
  - Capacity: $C = 1 - p$
- Capacity-Achieving Codes
  - A sequence of codes such that the
  - Probability of decoding failure tends to 0
  - Rate tends to capacity C
- Complexity vs. Gap to Capacity
  - For any $\varepsilon > 0$, what is the complexity of achieving a rate $(1-\varepsilon)C$?
  - Bounded complexity implies the complexity is bounded as $\varepsilon \to 0$

Low-Density Parity-Check (LDPC) Codes (1)

- Definition
  - Binary linear codes defined over the Galois field $\{0, 1\}$
  - Specified by the $m \times n$ parity-check (PC) matrix H
  - Each codeword x satisfies $Hx = 0$
  - The block length is n and the code rate is $\geq 1 - \frac{m}{n}$
  - Low-density implies the # of 1's in H is proportional to n
- Graphical Representation
  - The matrix H defines an undirected bipartite graph
  - Each column is associated with a bit in the codeword
  - Each row is associated with a parity-check equation
  - Each 1 in H defines an edge between a bit and check node

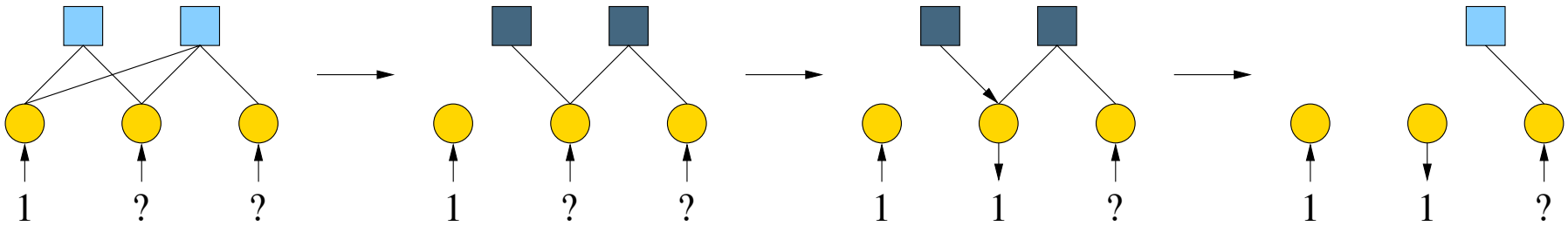
Low-Density Parity-Check (LDPC) Codes (2)



- Irregular Ensembles Defined by Degree Distribution (d.d.)
  - L_i (resp. R_i) is the fraction of bit (resp. check) nodes with degree i
  - λ_i (resp. ρ_i) is the fraction of edges with bit (resp. check) degree i
  - Associated functions: $L(x) = \sum_i L_i x^i$ and $\lambda(x) = \sum_i \lambda_i x^{i-1}$
- Random Permutation Between Bit and Check Nodes
  - Analysis averages over all possible permutations
  - Many results hold for almost all permutations as $n \to \infty$

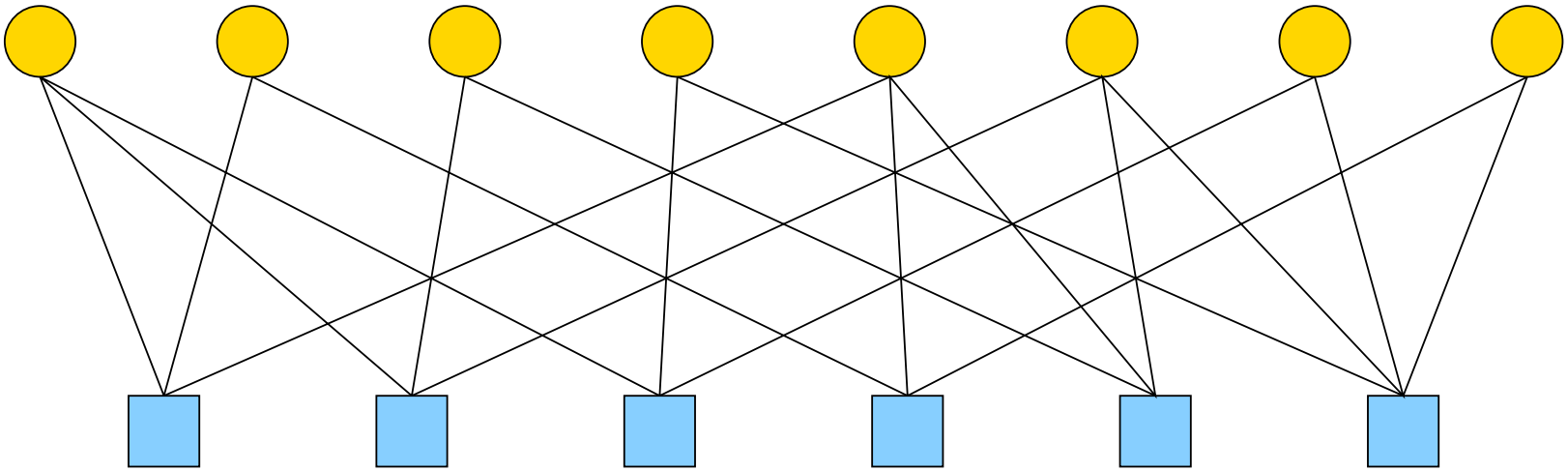
One Pass (Luby Style) Decoding for the BEC

- Constraint Nodes
  - Bit nodes assert that all edges have same value
  - Check nodes assert that all edge values sum to 0 (1 if shaded)
- Update Rules
  - 1. If any bit is known: Update neighboring checks and delete edges
  - 2. If no bit is known: Use degree-1 check to determine a bit

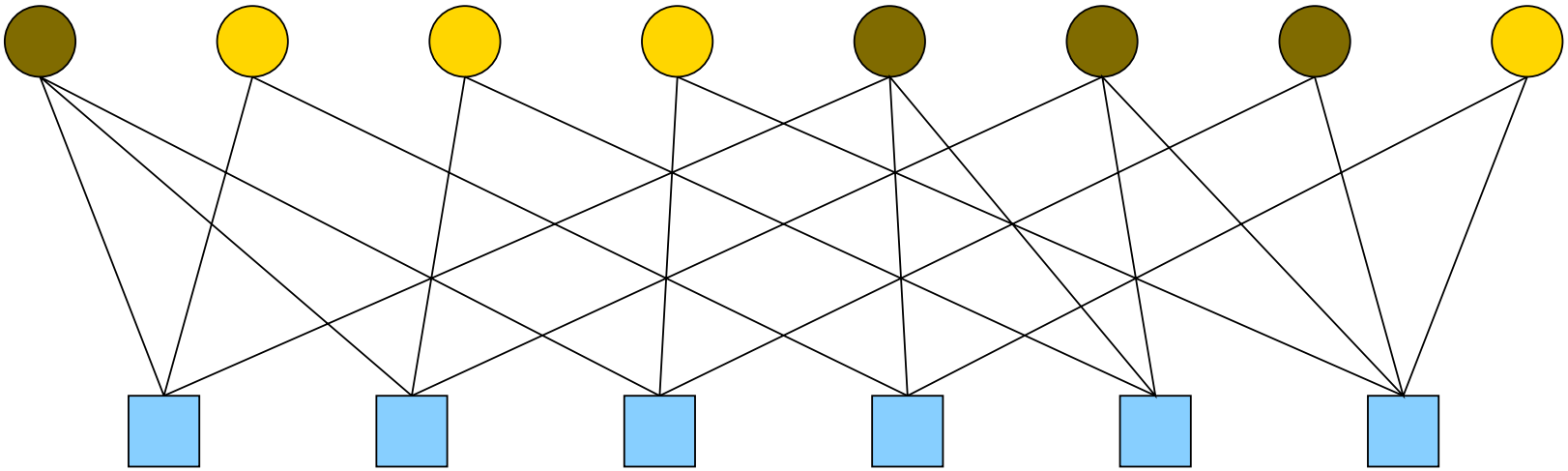


- Properties
  - Algorithm uses each edge at most once
  - Complexity $\sim$ number of edges in graph $= L'(1)n = R'(1)m$
  - Each erasure corrected uses one degree-1 check

Example: One Pass Decoding for All Zero Codeword

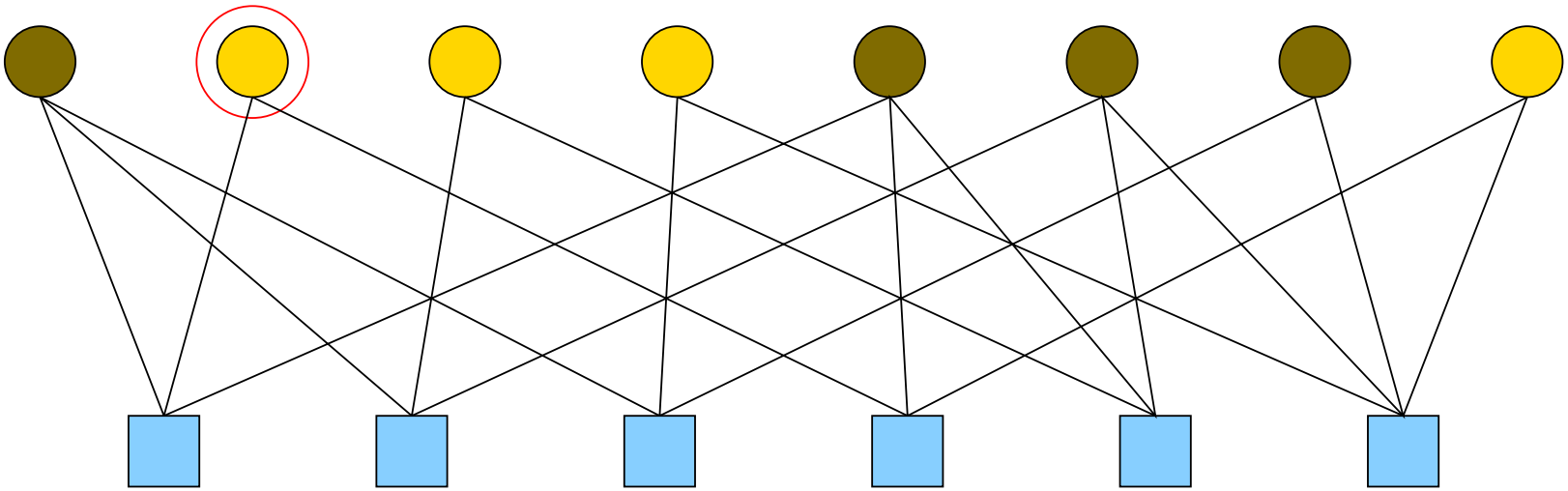


Example: One Pass Decoding for All Zero Codeword



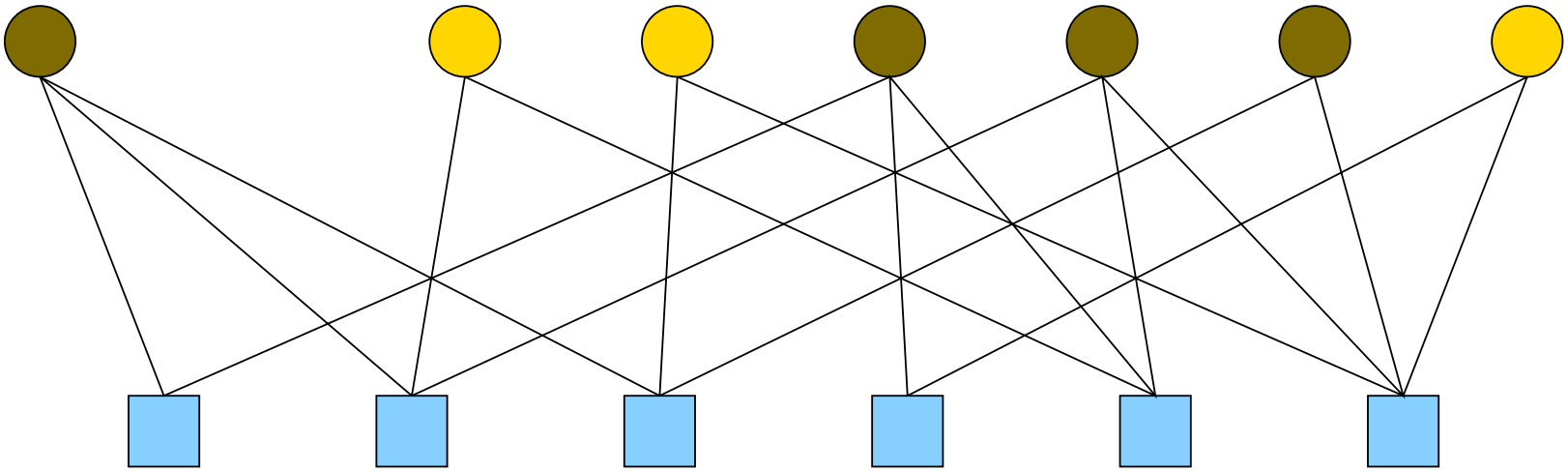
Add erasures from channel

Example: One Pass Decoding for All Zero Codeword



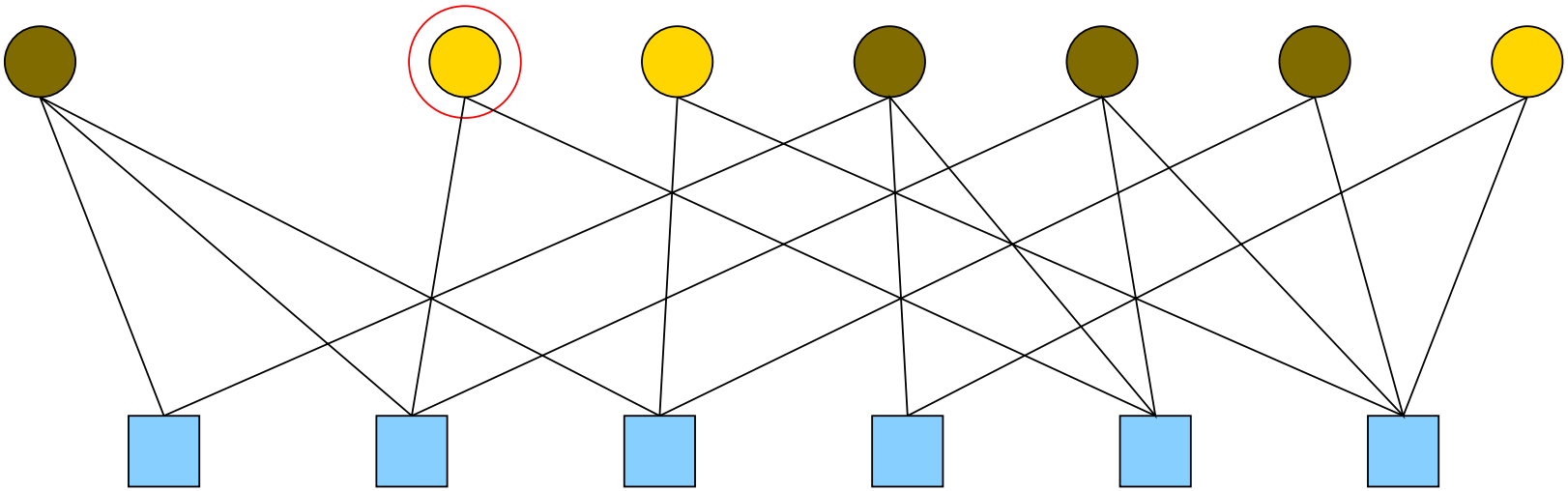
Remove known bits

Example: One Pass Decoding for All Zero Codeword



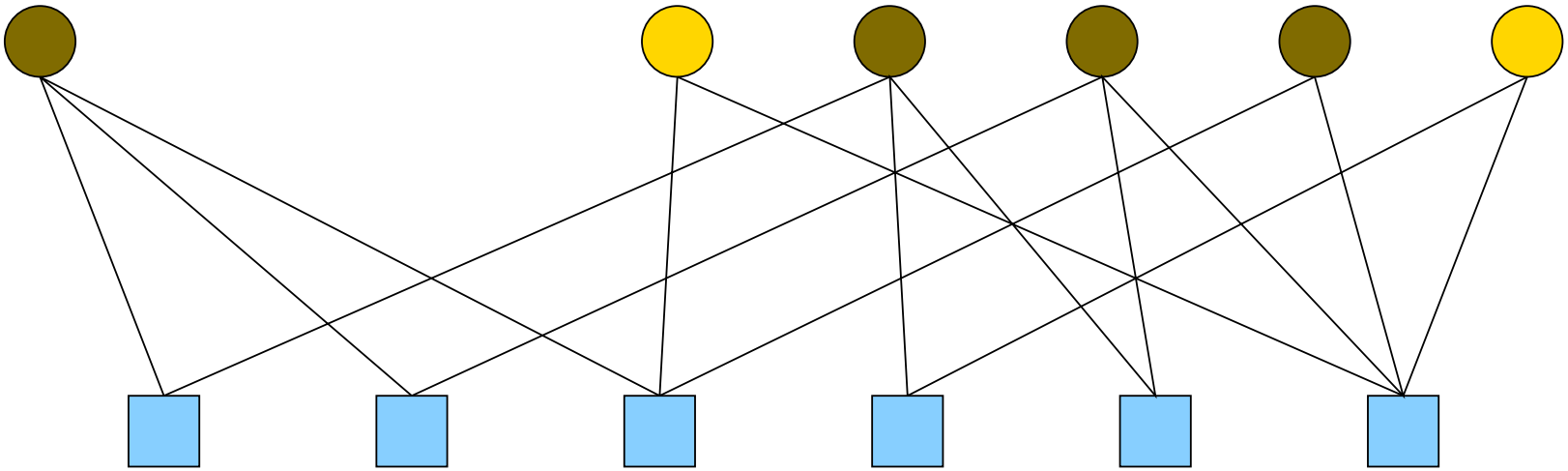
Remove known bits

Example: One Pass Decoding for All Zero Codeword



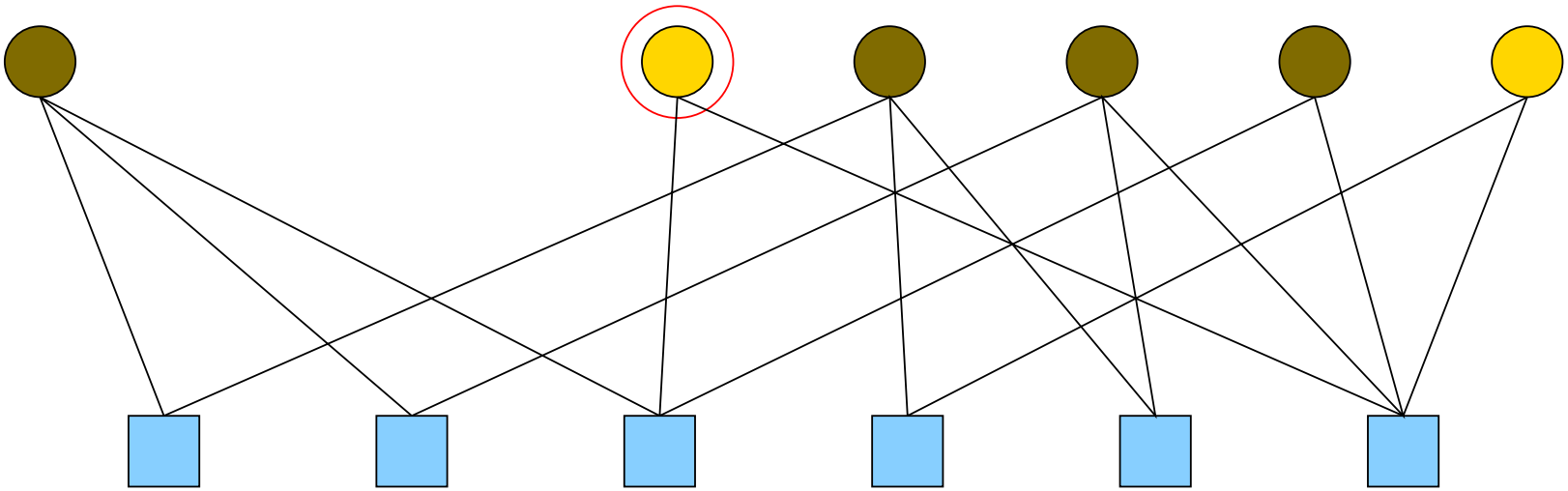
Remove known bits

Example: One Pass Decoding for All Zero Codeword



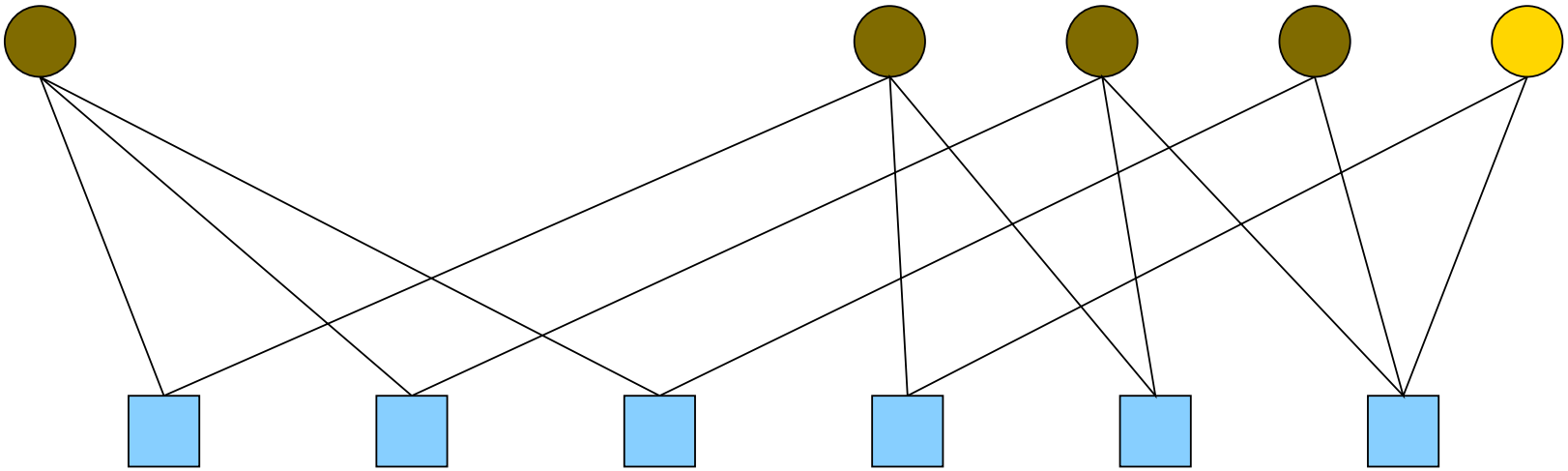
Remove known bits

Example: One Pass Decoding for All Zero Codeword



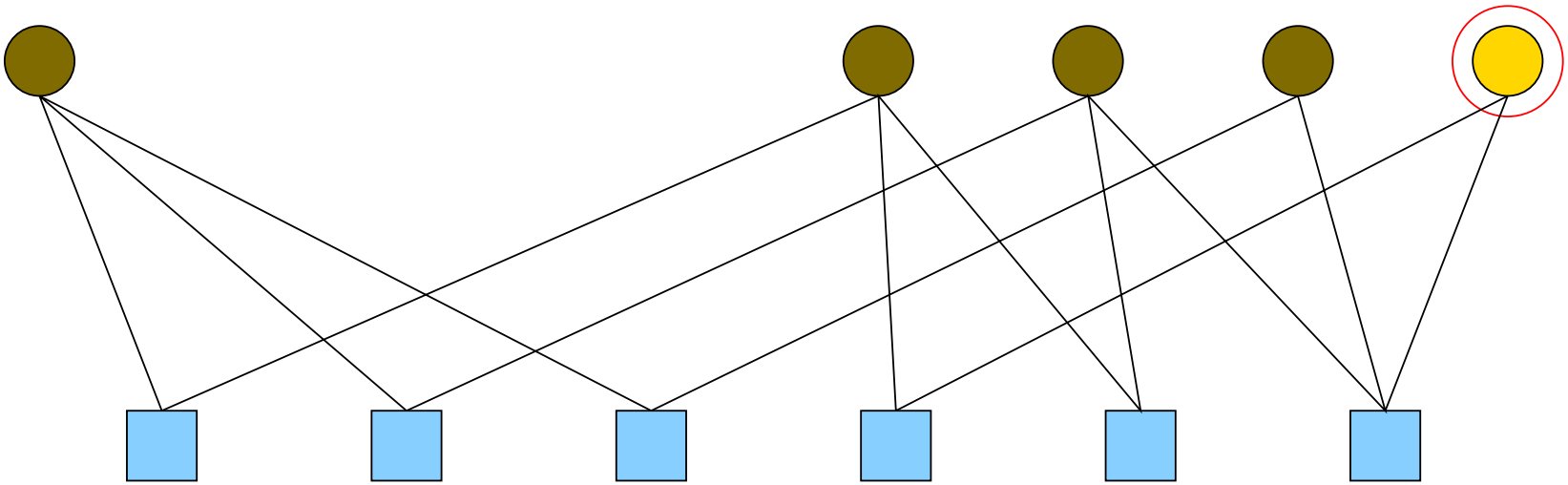
Remove known bits

Example: One Pass Decoding for All Zero Codeword



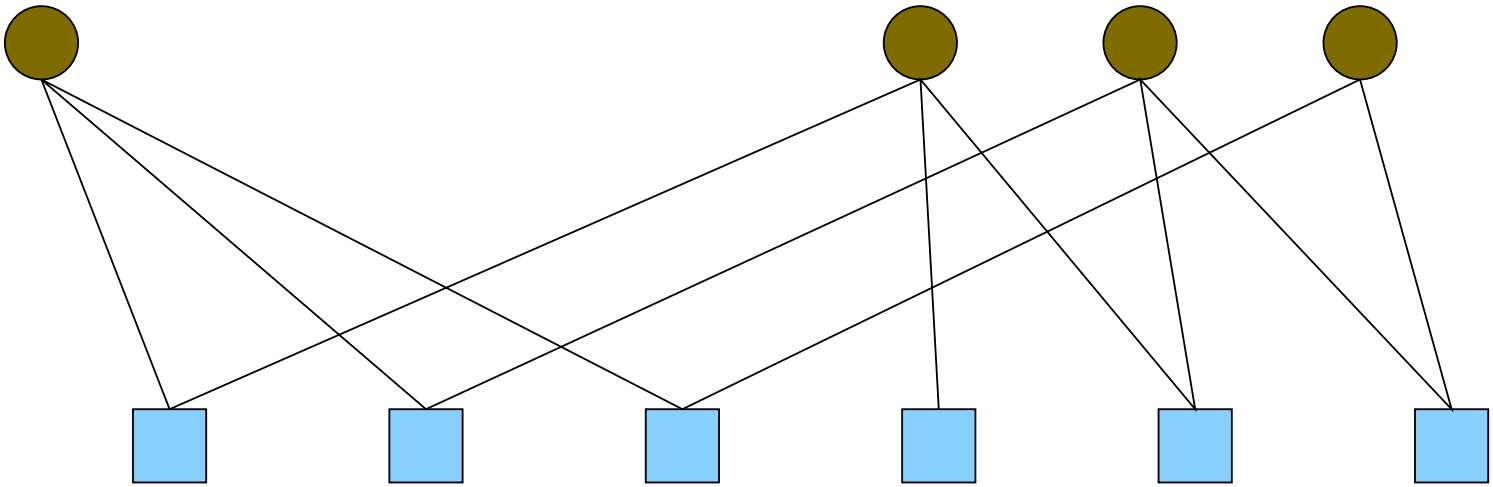
Remove known bits

Example: One Pass Decoding for All Zero Codeword

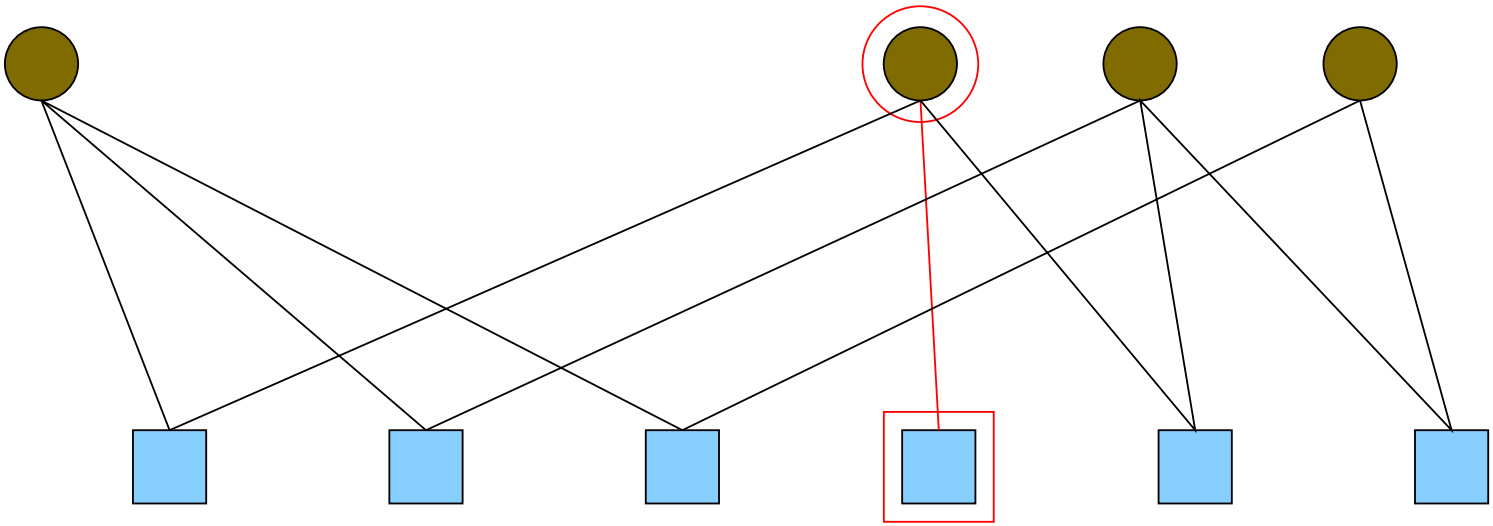


Remove known bits

Example: One Pass Decoding for All Zero Codeword

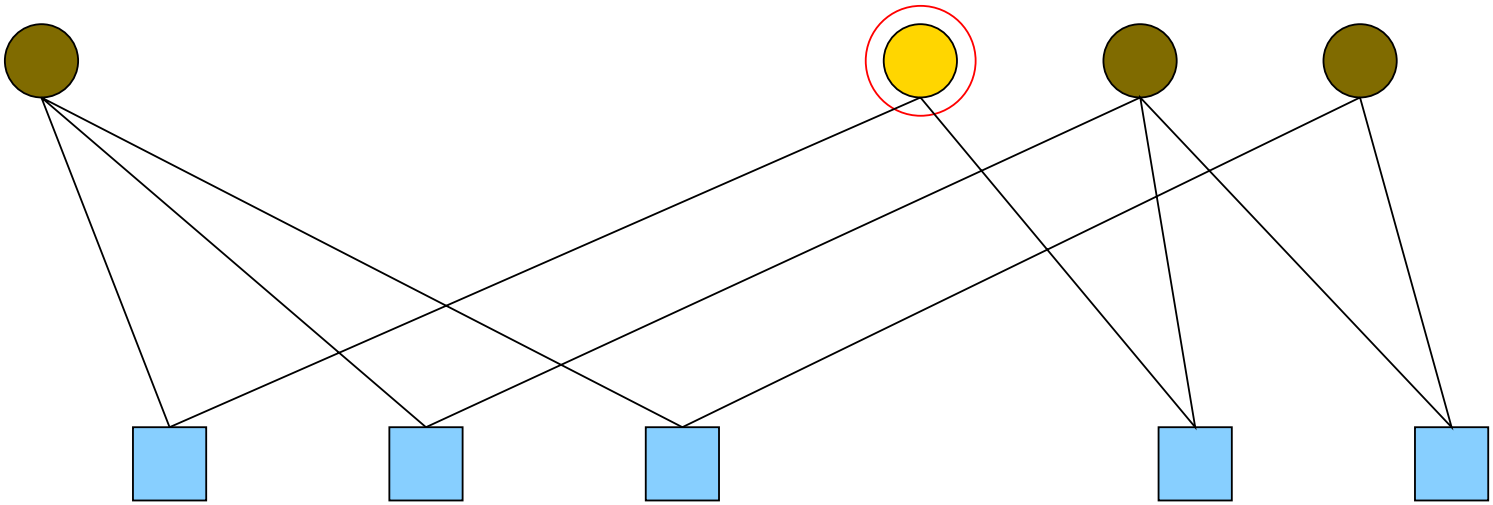


Example: One Pass Decoding for All Zero Codeword



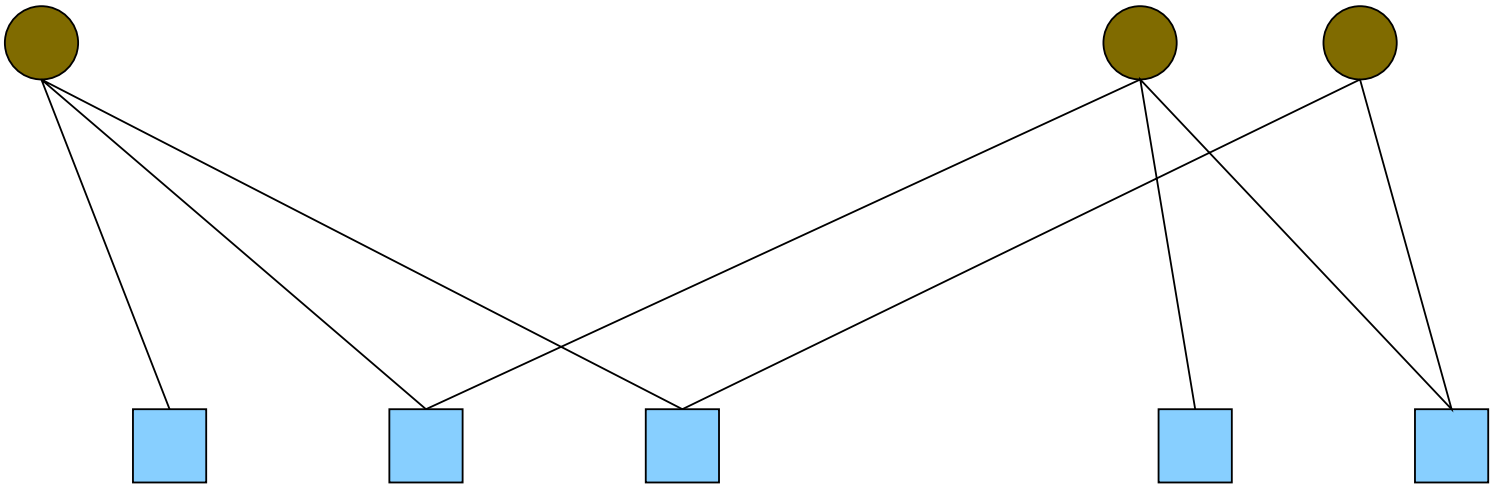
Use degree-1 check to determine a bit

Example: One Pass Decoding for All Zero Codeword

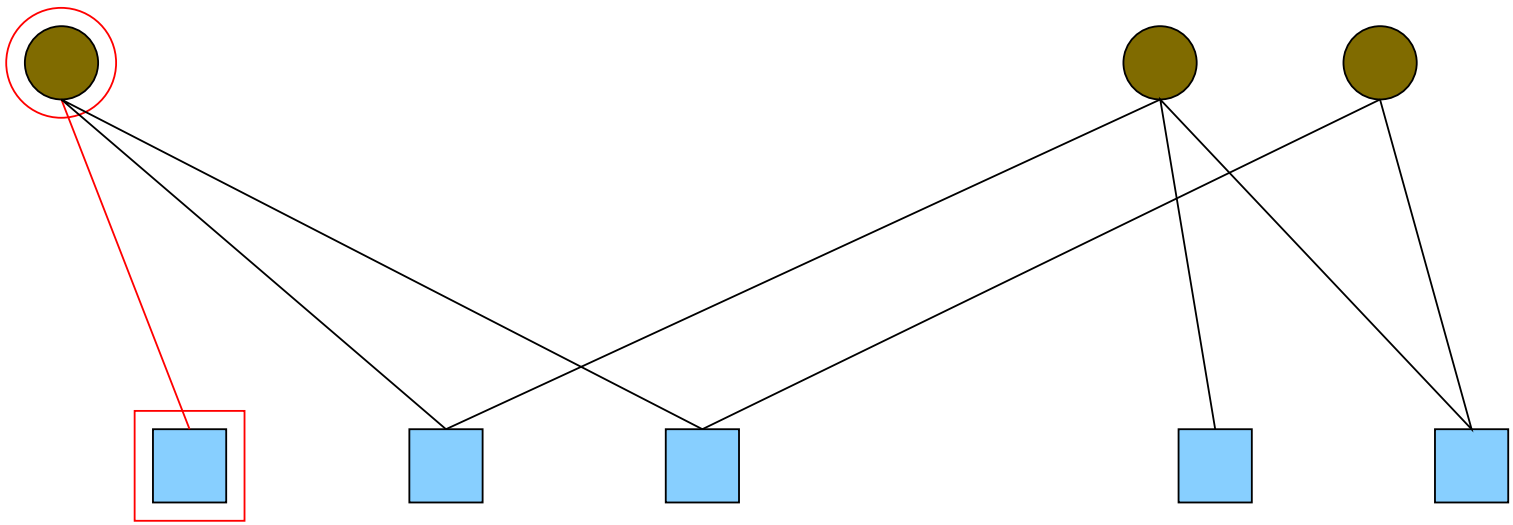


Remove known bit

Example: One Pass Decoding for All Zero Codeword

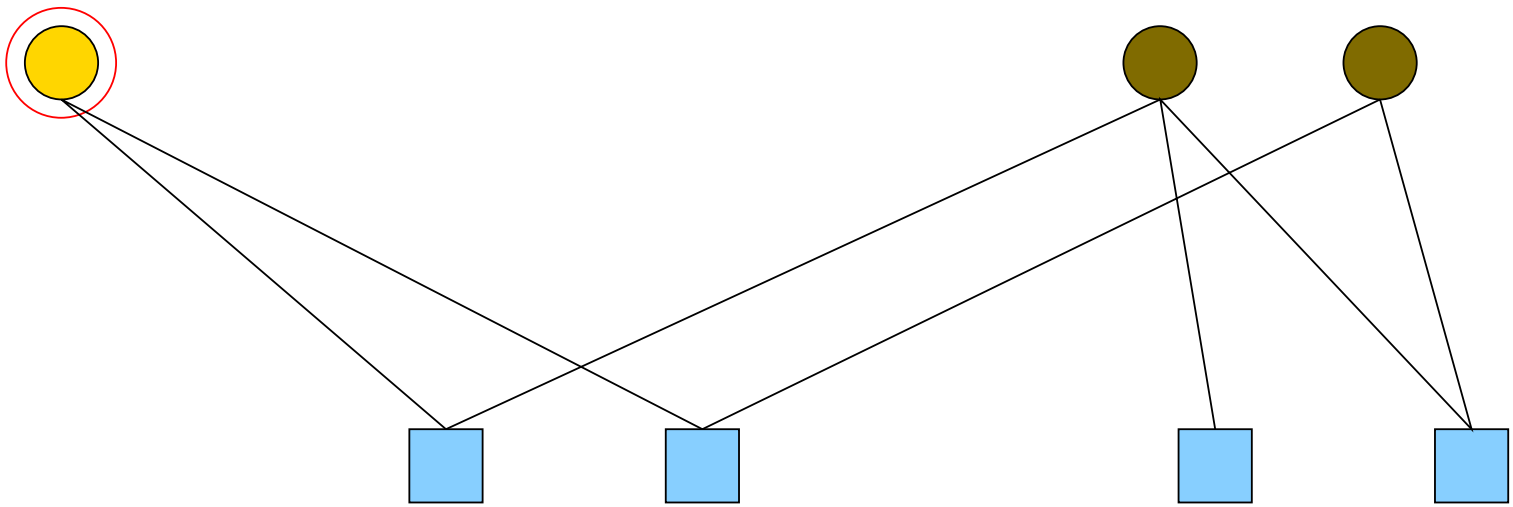


Example: One Pass Decoding for All Zero Codeword



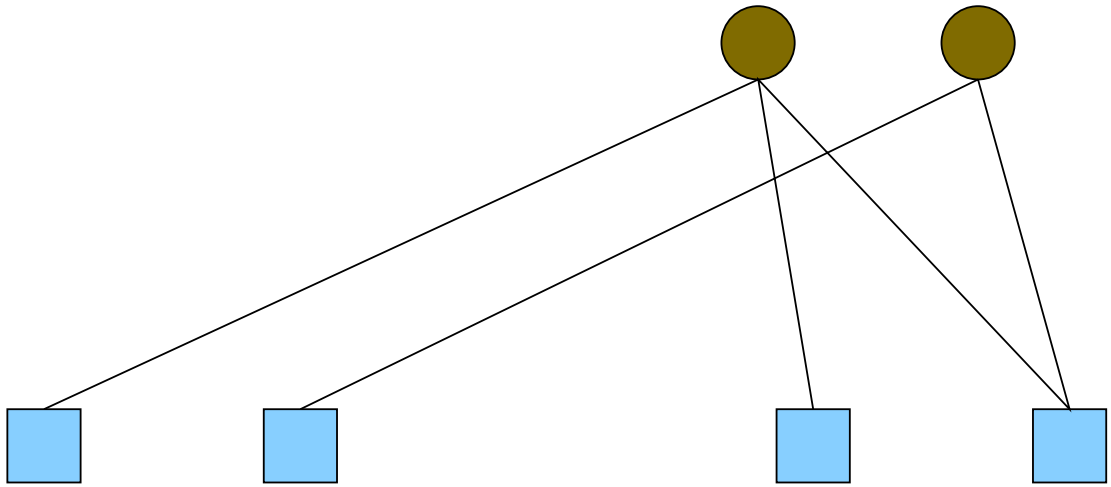
Use degree-1 check to determine a bit

Example: One Pass Decoding for All Zero Codeword

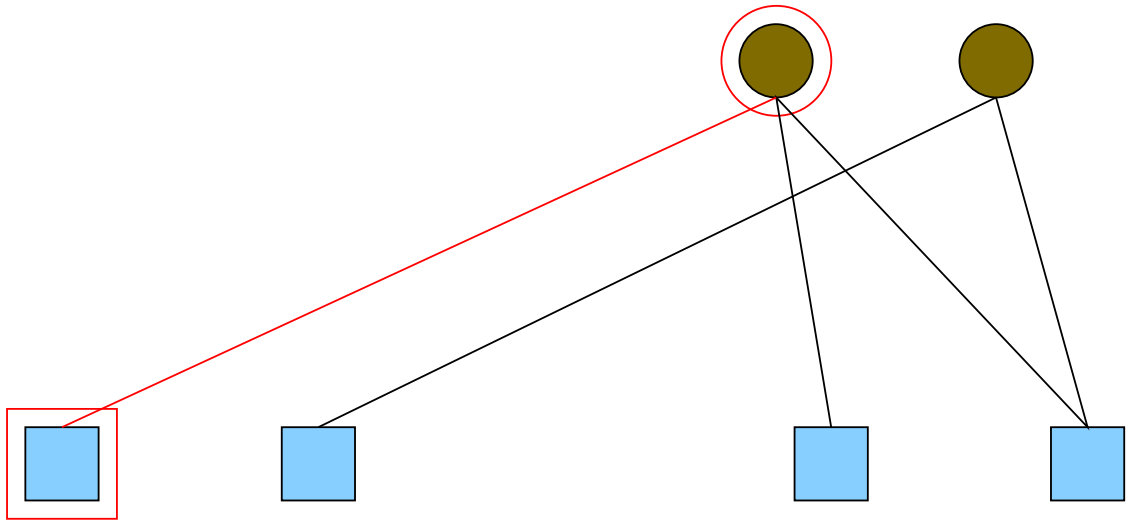


Remove known bit

Example: One Pass Decoding for All Zero Codeword

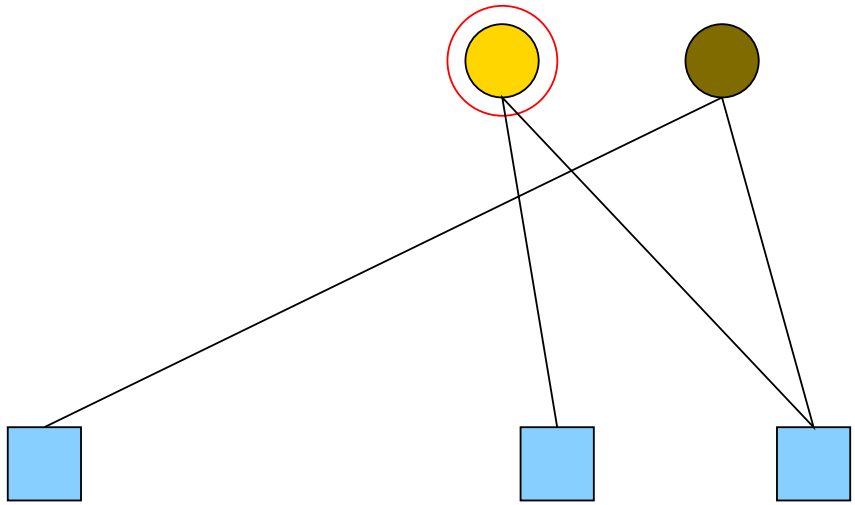


Example: One Pass Decoding for All Zero Codeword



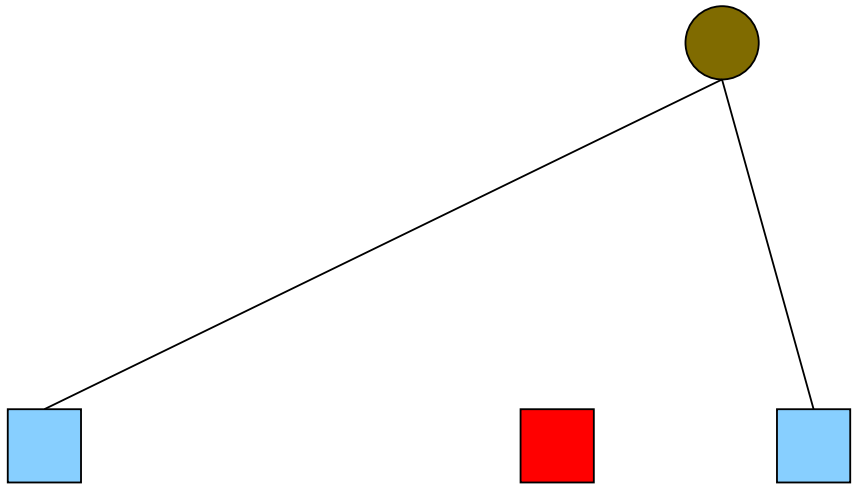
Use degree-1 check to determine a bit

Example: One Pass Decoding for All Zero Codeword



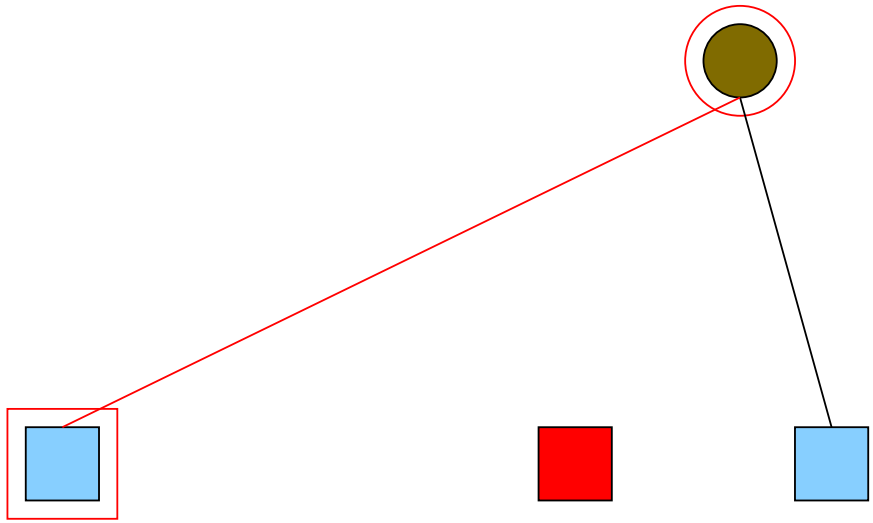
Remove known bit

Example: One Pass Decoding for All Zero Codeword



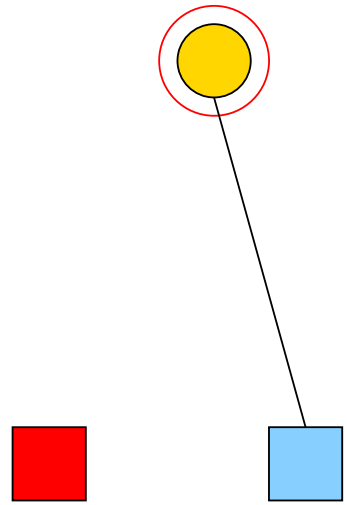
Red check lost all edges before determining a bit

Example: One Pass Decoding for All Zero Codeword



Use degree-1 check to determine a bit

Example: One Pass Decoding for All Zero Codeword



Remove known bit

Example: One Pass Decoding for All Zero Codeword

Notice the two remaining checks did not determine bits

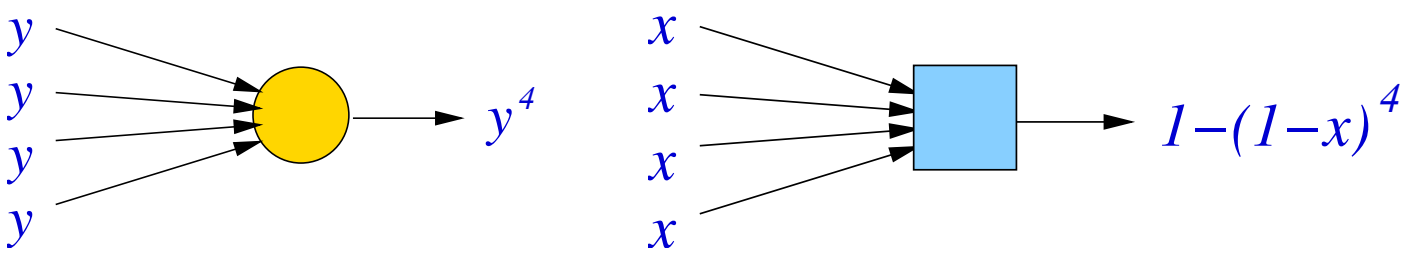
Check Loss and LDPC Codes

Check loss occurs when a parity-check node loses all of its edges without determining an erased bit

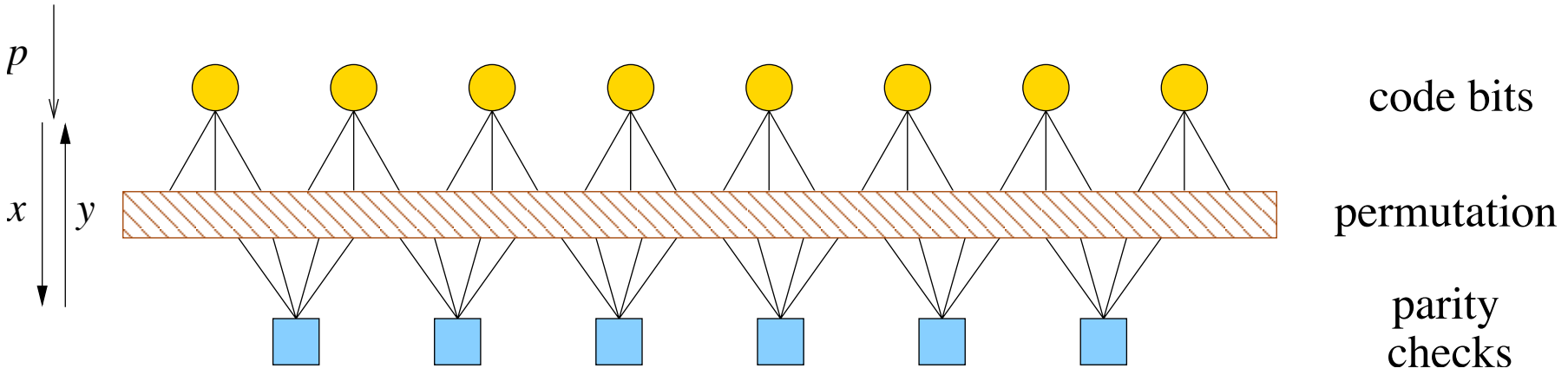
- The Problem With Check Loss
  - To achieve capacity, almost all checks must determine erased bits
  - Most check loss occurs during the initial channel observations
  - LDPC codes lose an initial fraction of $\sum_i R_i(1-p)^i = R(1-p)$
  - So the max # of correctable erasures is $p^* n \leq m - mR(1-p^*)$
- This Simple Bound
  - Holds even for maximum likelihood decoding
  - Is identical to Shokrollahi's 1999 bound for iterative decoding
  - Matches or beats previous information theoretic bounds
  - Can be extended to general channels

Iterative Message-Passing Decoding

- Iterative Decoding
  - Each iteration has two phases
  - 1st Phase: Each bit passes a message to each adjacent check
  - 2nd Phase: Each check passes a message to each adjacent bit
  - Each output message depends only on the input messages
- Iterative Decoding for the BEC
  - All messages are either the correct value or an erasure
  - Bits pass the correct value unless all other inputs are erased
  - Checks pass the correct value only if all other inputs are correct
  - Let x/y be the erasure prob. of bit/check output messages



Density Evolution (DE) for the BEC



- Track the avg. fraction of erasure messages vs. iteration
  - Messages are treated as independent (no short cycles for large n)
  - If it converges to 0, then decoder succeeds almost surely for large n
- The lth Iteration of Decoding for an LDPC Code
  - Let $x^{(l)}/y^{(l)}$ be the erasure prob. of bit/check output messages
  - Avg. over the check d.d. gives: $y^{(l+1)} = 1 - \rho(1 - x^{(l)})$
  - Avg. over the bit d.d. gives: $x^{(l+1)} = p\,\lambda\left(y^{(l+1)}\right)$
- 1D recursion: $x^{(l+1)} = p\,\lambda\left(1 - \rho(1 - x^{(l)})\right)$

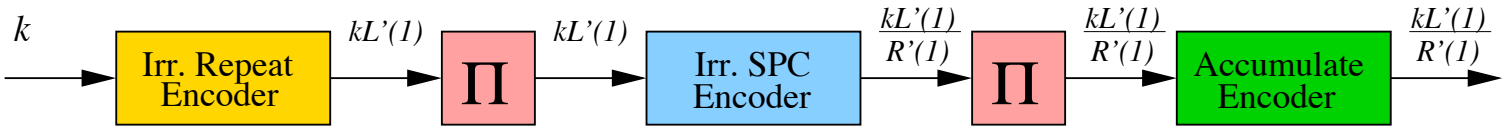
Capacity-Achieving LDPC Codes

Use the DE fixed point equation to solve for c.a. codes

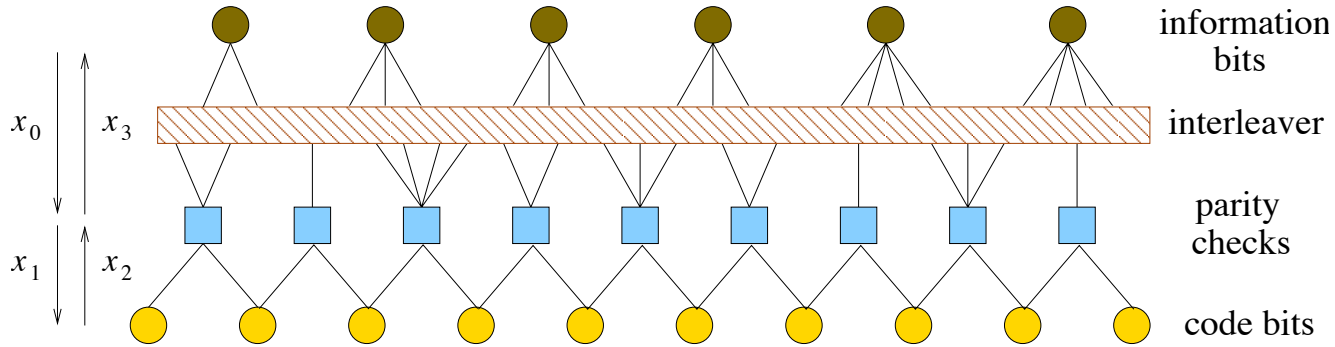
$$x = p\,\lambda\left(1-\rho(1-x)\right) \quad \Longrightarrow \quad \lambda(x) = \frac{1}{p}\left(1-\rho^{-1}(1-x)\right)$$

1. Choose sequence of check distributions: $\rho^{(k)}(x) = x^k$
2. Solve for sequence of bit distributions: $\lambda^{(k)}(x) = \left(1-(1-x)^{1/k}\right)$
3. Verify that the power series expansion of $\lambda^{(k)}(x)$ is non-negative
4. Truncate power series of $\lambda^{(k)}(x)$ so that: $\overline{\lambda}^{(k)}(1) = 1$
5. Truncation implies: $p\,\overline{\lambda}^{(k)}\left(1-\rho^{(k)}(1-x)\right) < x$ for $x \in [0,1]$
6. DE converges and decoding succeeds almost surely for large n
7. Complexity (edges per info bit) $\sim \frac{pk}{1-p}$ is unbounded as $k \to \infty$

Repeat-Accumulate (RA) Type Codes

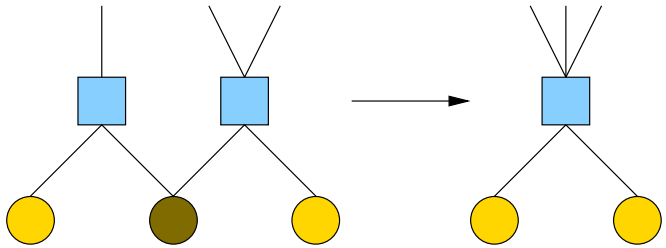


- IRA Codes: Encoding Point of View
  - Irregular Repeat: fraction L_i of bits repeated i times
  - Irregular Single Parity: fraction R_i checks have degree i
  - Accumulate mapping: $x_1^n \rightarrow y_1^n$ with $y_i = y_{i-1} + x_i$
  - Fraction α of the information bits are sent
- IRA Codes: Decoding Point of View
  - LDPC type decoding graph with added "accumulate" section
  - Fraction of checks lost $= (1-p)^2 R$ $(\alpha(1-p))$



Graph Reduction for IRA Codes (1)

- Graph Reduction $\mathcal{G}_{\text{IRA}}$
  - Each erased code bit is removed by summing two PC equations
  - Summing two PC equations causes their degrees to add
  - Clearly, this will modify the check d.d.



- What is left?
  - After all erased code bits are gone
  - Remaining "known" code bits are absorbed into checks
  - The residual graph then defines an LDPC code
  - So $\mathcal{G}_{\text{IRA}}$ maps an (L, R) IRA code into an $(L, \widetilde{R})$ LDPC code

Graph Reduction for IRA Codes (2)

The New Degree Distribution $\widetilde{R}(x)$

- Check degrees are i.i.d. with generating function (g.f.) $R(x)$
- Removing i erasures in a row causes $i+1$ degrees to be summed
- g.f. of the sum is $R(x)^{i+1}$ and the runlength prob is $(1-p)\,p^i$, so

$$\widetilde{R}(x) = \sum (1-p)p^i R(x)^{i+1} = \frac{(1-p)R(x)}{1-p\,R(x)}$$

Density Evolution via Graph Reduction

- Apply DE to reduced LDPC code: $x^{(l+1)} = p\,\lambda\left(1-\widetilde{\rho}(1-x^{(l)})\right)$
- Using $\widetilde{\rho}(x) = \widetilde{R}'(x)/\widetilde{R}'(1)$ gives

$$x^{(l+1)} = (1-\alpha(1-p))\,\lambda\left(1-\frac{(1-p)^2\rho(x^{(l)})}{(1-p\,R(x^{(l)}))^2}\right)$$

Complexity vs. Gap to Capacity

What is the complexity of achieving a rate $(1-\varepsilon)C$?

Theorem (Sason&Urbanke A)

Under iterative message-passing decoding, the decoding complexity per information bit of LDPC codes, without puncturing, grows at least like $\log \frac{1}{\varepsilon}$ *(i.e., the log of the inverse of the gap to capacity).*

Theorem (Sason&Urbanke B)

Under iterative message-passing decoding, the decoding complexity per information bit of systematic IRA (SIRA) codes grows at least like $\log \frac{1}{\varepsilon}$ *(i.e., the log of the inverse of the gap to capacity).*

Decoding complexity is *unbounded* as the gap to capacity vanishes!

Capacity-Achieving IRA Codes

Use the DE fixed point equation ($\alpha = 0$) to solve for c.a. codes

$$x = \lambda\left(1 - \frac{(1-p)^2\rho(x)}{(1-p\,R(x))^2}\right) \implies \lambda^{-1}(x) = 1 - \frac{(1-p)^2\rho(x)}{(1-p\,R(x))^2}$$

1. Choose check regular with degree 3: $R(x) = x^3$ and $\rho(x) = x^2$
2. Bit distribution solved by root of 3rd degree polynomial

$$\lambda(x) = 1 - \sqrt{\frac{4(1-p)}{3p\sqrt{1-x}}}\sinh\left(\frac{1}{3}\sinh^{-1}\left(\sqrt{\frac{27p(1-x)^{3/2}}{4(1-p)^3}}\right)\right) \quad (1)$$

3. Verify that the power series expansion of $\lambda(x)$ is non-negative
4. Truncate $\lambda(x)$ by converting large degree bits to pilot bits
5. This truncation implies DE converges to 0
6. Complexity (edges per info bit) $\sim \frac{5}{1-p}$ is bounded!

Polya's Criterion and Non-Negativity Proofs (1)

Motivation

Polya's Criterion gives a sufficient condition for a function $F(t)$ to be the characteristic function of a random variable.

Theorem (Polya's Criterion)

If F is real, symmetric, non-negative, and convex non-increasing on $[0, \infty)$, then its Fourier transform is non-negative.

Theorem (A Discrete Polya Criterion)

If F is real, symmetric, 2π-periodic, and convex non-increasing on $(0, \pi]$, then its Fourier series has non-negative coefficients.

Polya's Criterion and Non-Negativity Proofs (2)

Corollary (Non-negative Power Series)

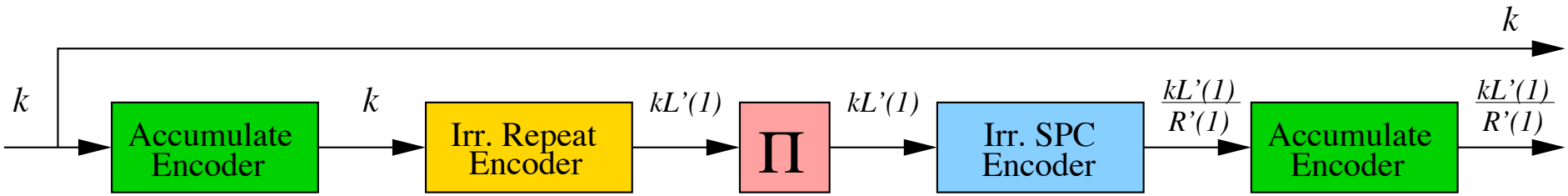
Let the function $g : \mathbb{C} \to \mathbb{C}$ be analytic on the unit disc except possibly $x = 1$. If the real function $h(x) = Re\left\{g\left(e^{ix}\right)\right\}$ is symmetric, convex for $x \in [0, \pi]$, and satisfies $\int_0^\pi h(x)dx \geq 0$, then g has a power series expansion about zero with non-negative coefficients.

Application

Consider the bit d.d. $\lambda(x)$ of the check-regular non-systematic IRA ensemble. This d.d. has a non-negative power series expansion for $p \in (0, 1)$. To prove this, one can numerically verify the convexity using complex interval arithmetic and then apply the Corollary.

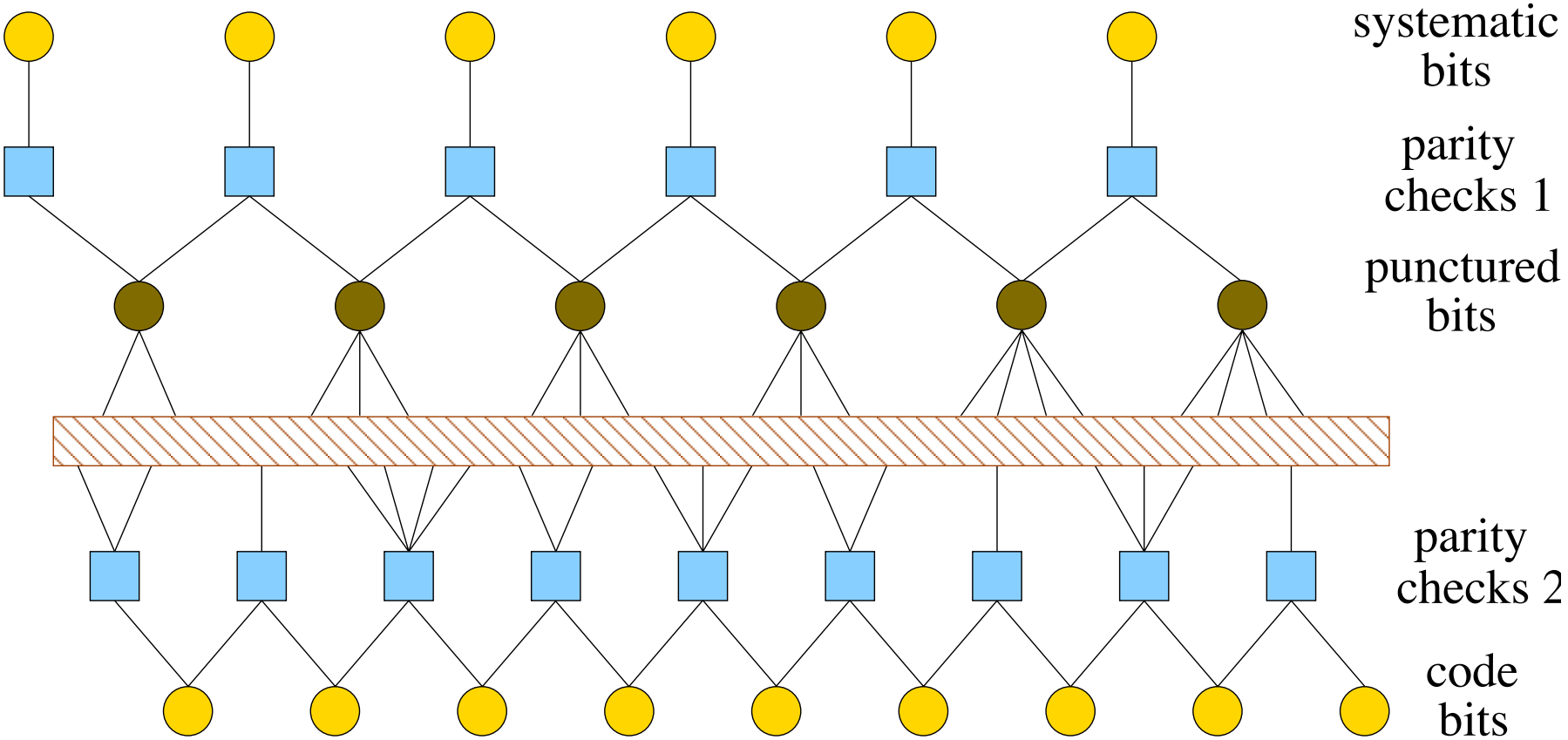
Accumulate-Repeat-Accumulate (ARA) Codes

- These codes are a generalization of the IRA codes; they were introduced by Abbasfar, Divsalar and Yao (ISIT 2004)
- They have good performance and simple linear-time encoding



- Encoder diagram for the systematic ARA ensemble
  - Accumulate block is the rate-1 $\frac{1}{1+D}$ encoder
  - Irregular Repeat: fraction L_i of bits repeated i times
  - Irregular SPC: fraction R_i single parity checks have degree i
  - Block sizes are shown starting with k info bits

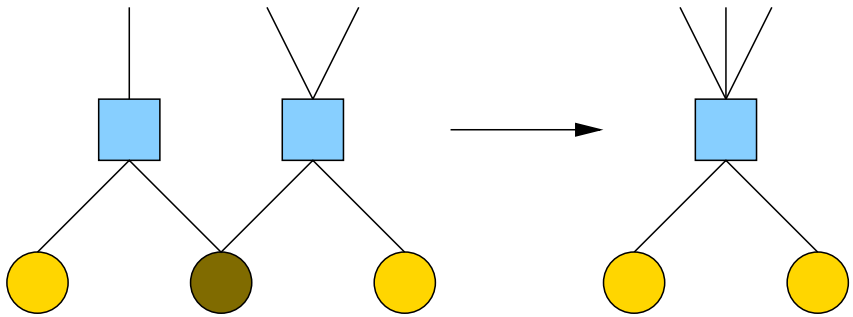
The Decoding Graph for ARA Codes



- Shading is used to denote punctured or erased bits

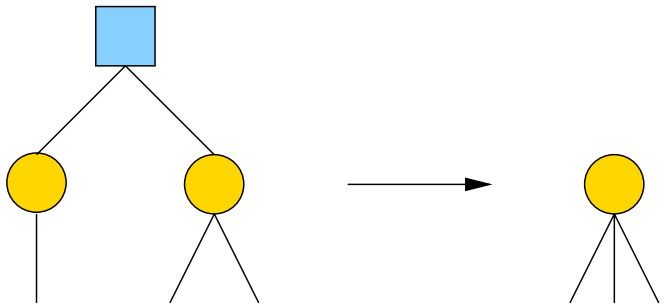
Graph Reduction for Code Bits

- Any “code bit" node whose value is not erased by the BEC can be removed from the graph by absorbing its value into its two “parity-check 2" nodes.
- When the value of a “code bit" node is erased, one can merge the two “parity-check 2" nodes which are connected to it (by summing the equations) and this removes the “code bit" from the graph.
- Merging two “parity-check 2" nodes causes their degrees to be summed.

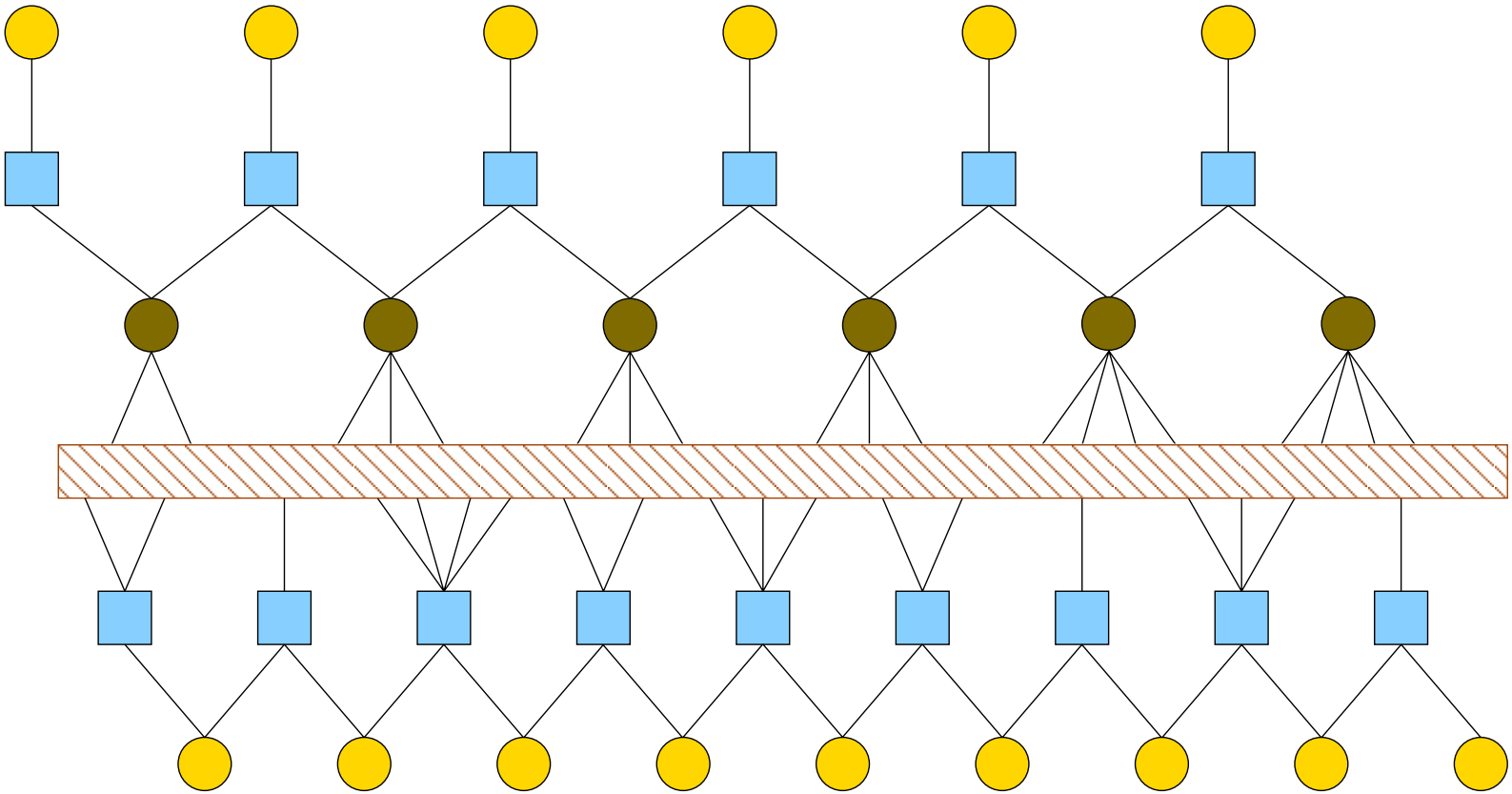


Graph Reduction for Systematic Bits

- The “systematic bit" nodes in the Tanner graph of the systematic ARA codes only provide channel information. Erasures make them worthless, and they can be removed along with their “parity-check 1" nodes without affecting the decoder.
- When the value of a “systematic bit" node is observed (assume the value is zero w.o.l.o.g.), it can be removed leaving a degree 2 parity-check.
- Degree 2 parity-checks imply equality, and allow the connected “punctured bit" nodes to be merged (summing their degrees).

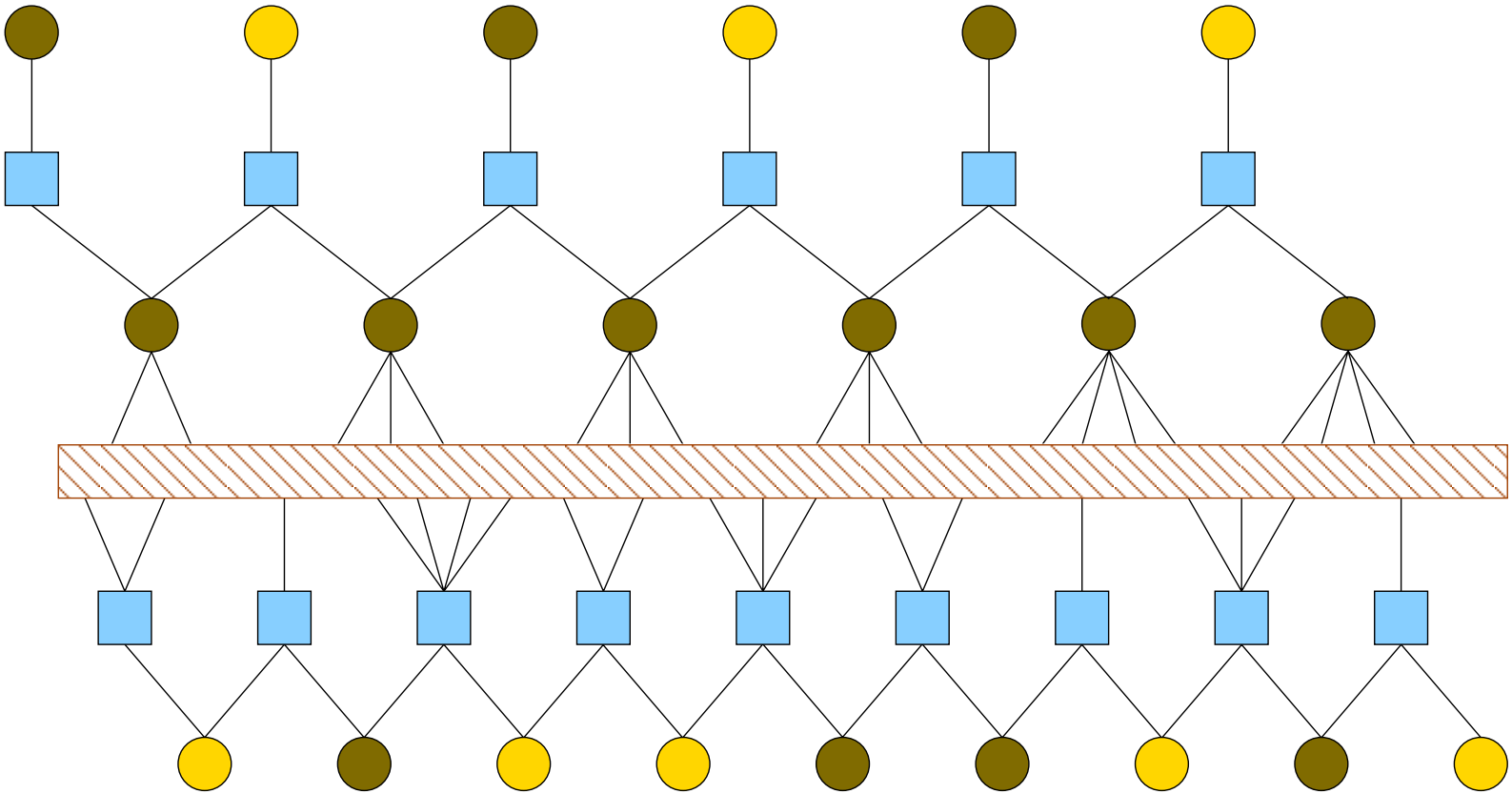


Example of Graph Reduction



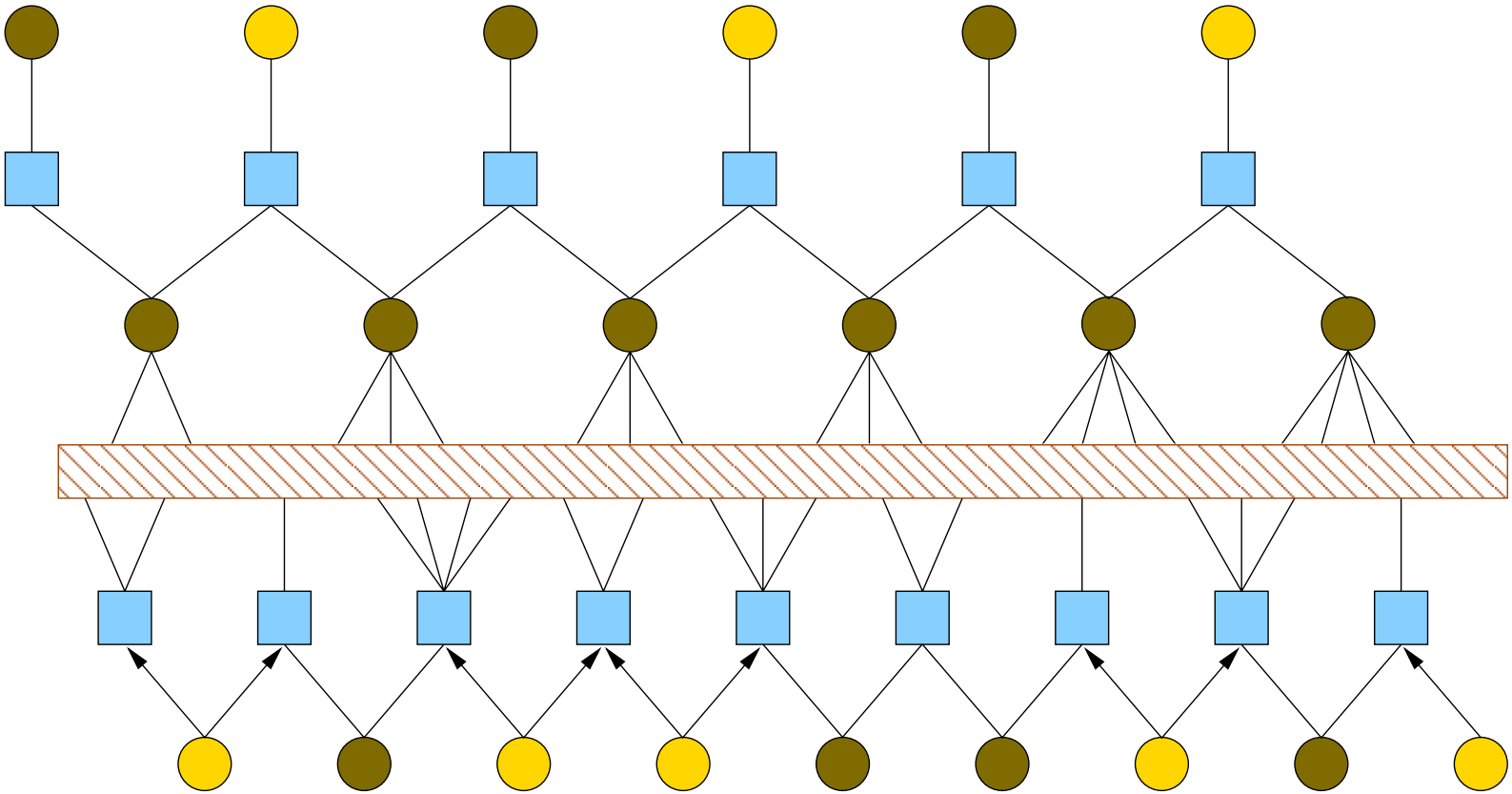
Original Tanner graph

Example of Graph Reduction



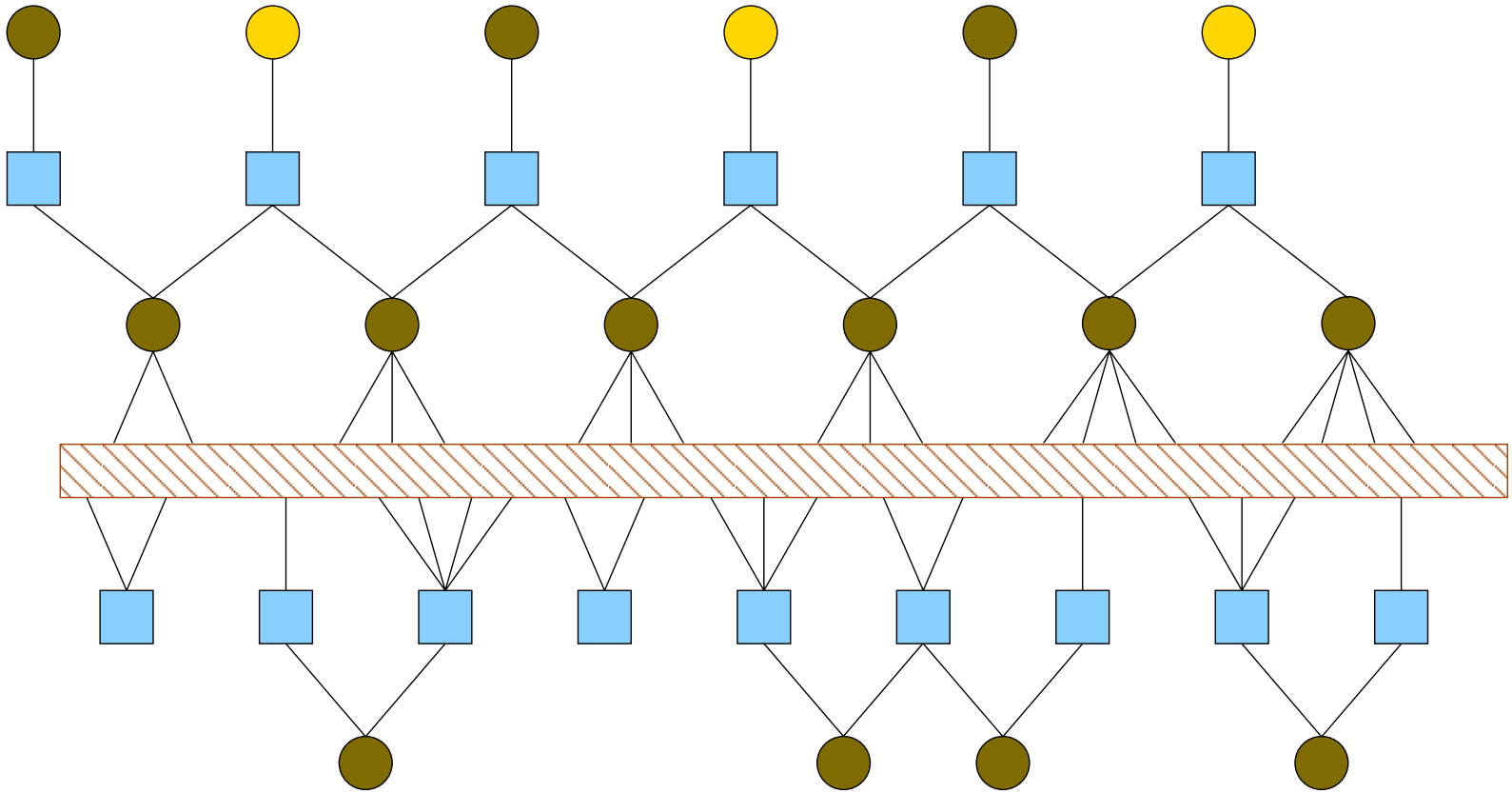
Add erasures from channel

Example of Graph Reduction



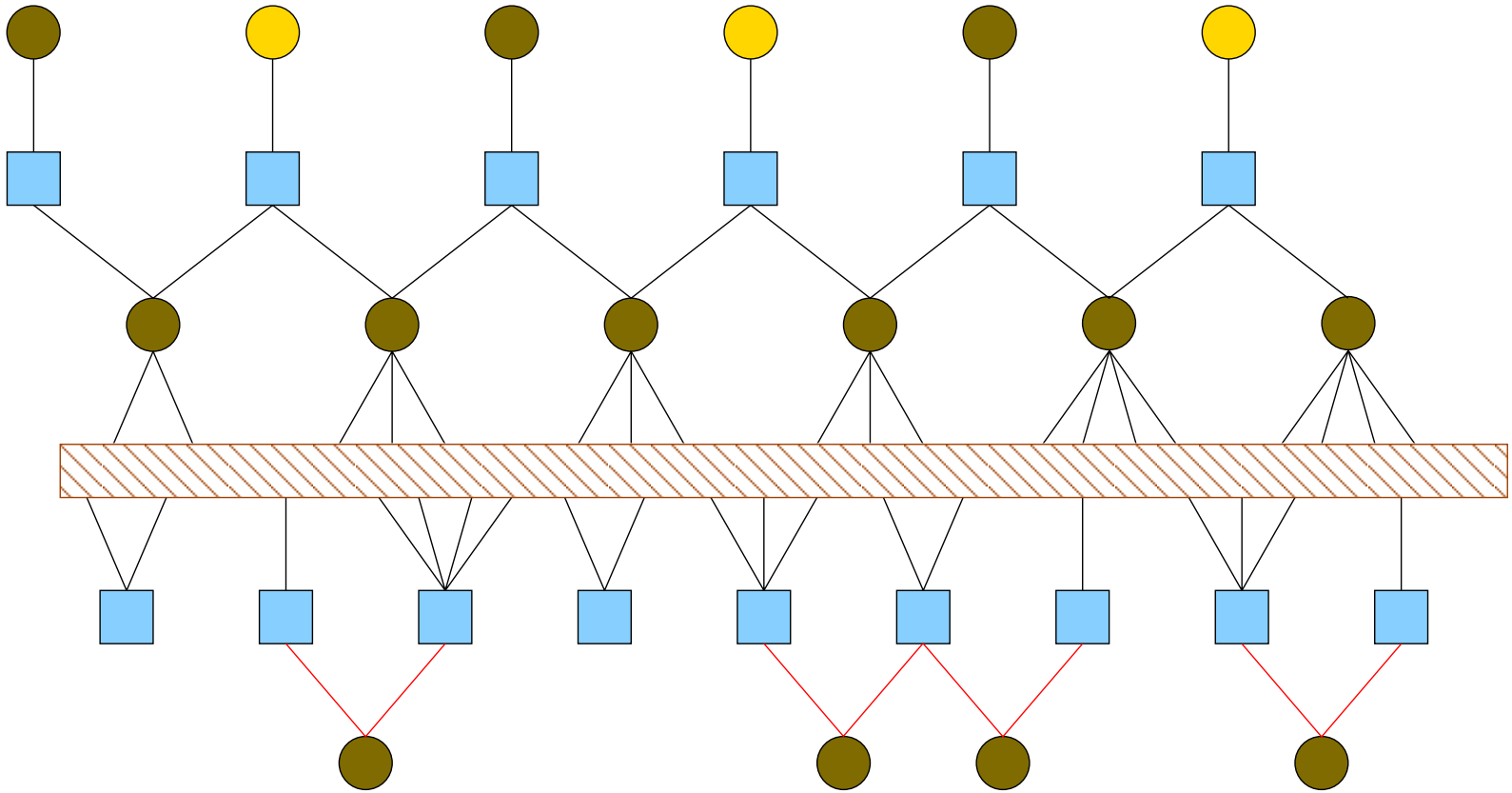
Mark known code bits

Example of Graph Reduction



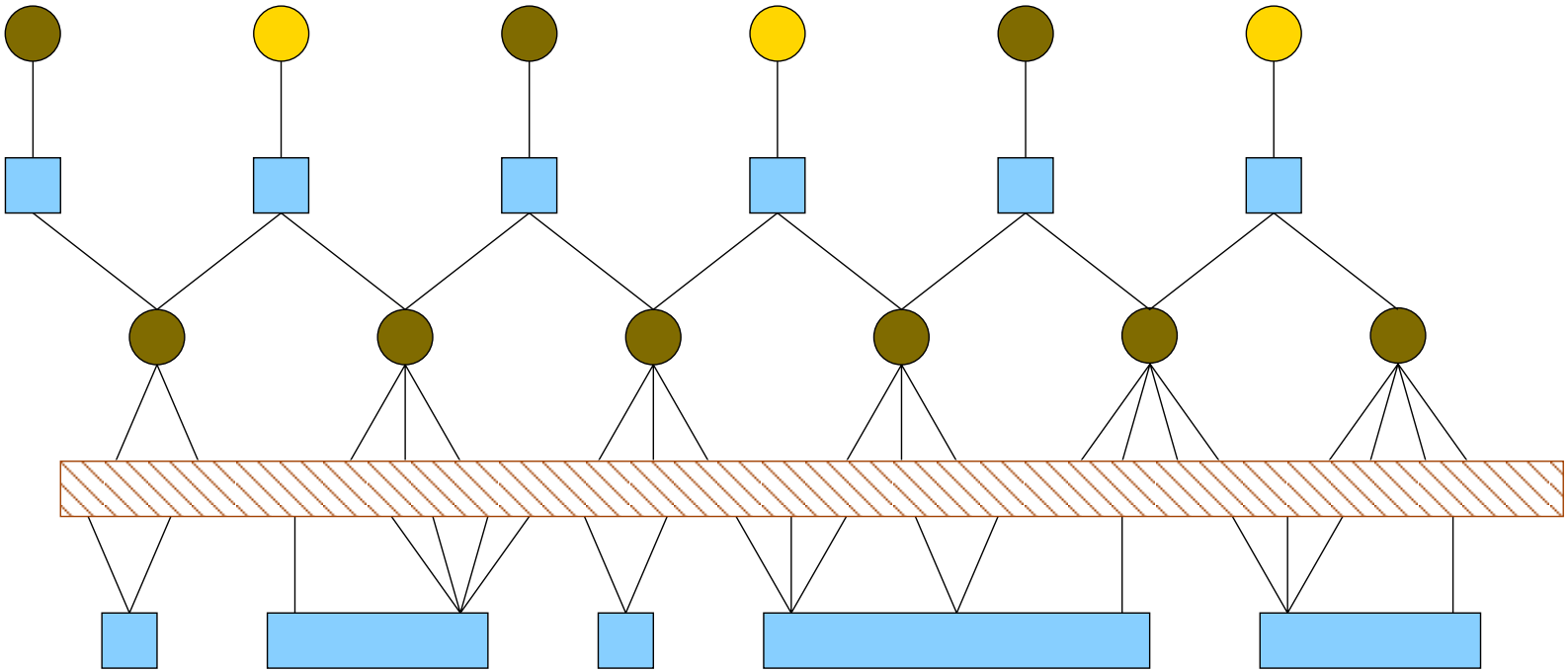
Merge values into checks

Example of Graph Reduction



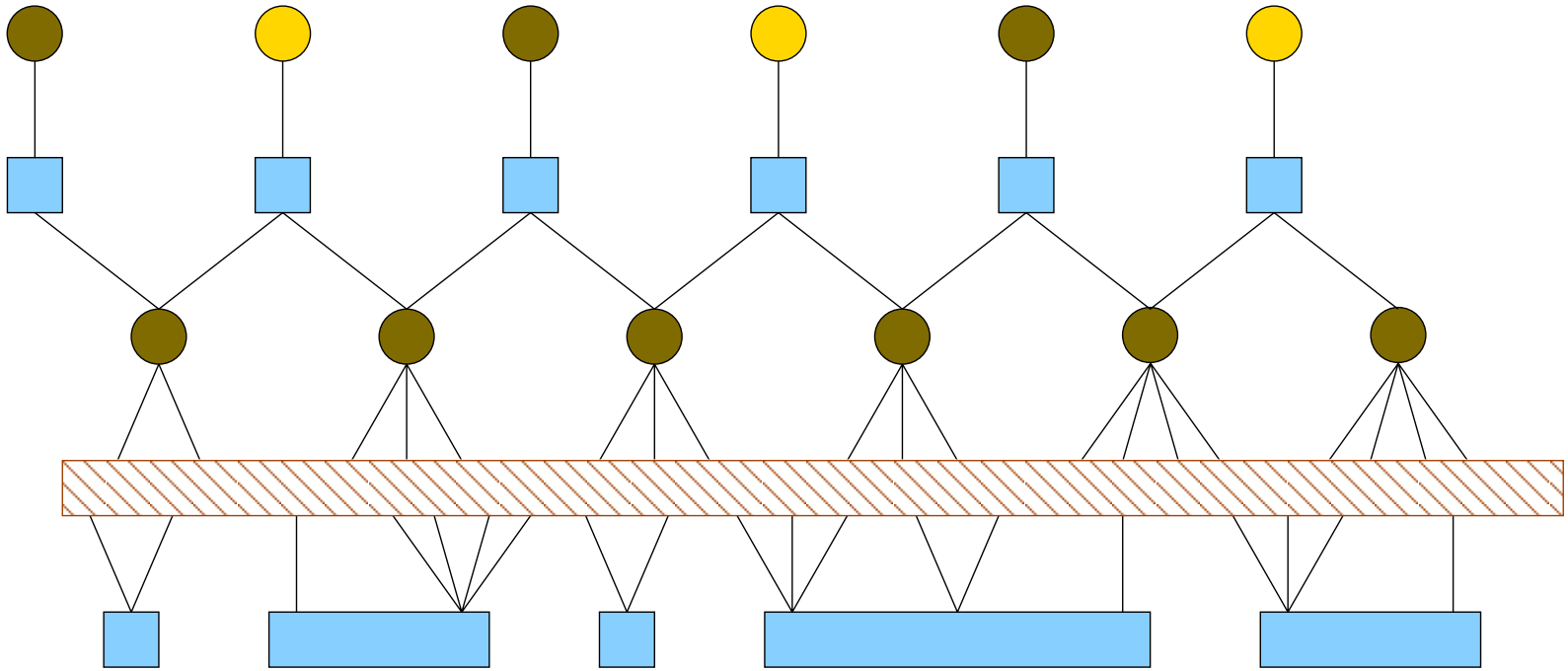
Mark unknown code bits

Example of Graph Reduction



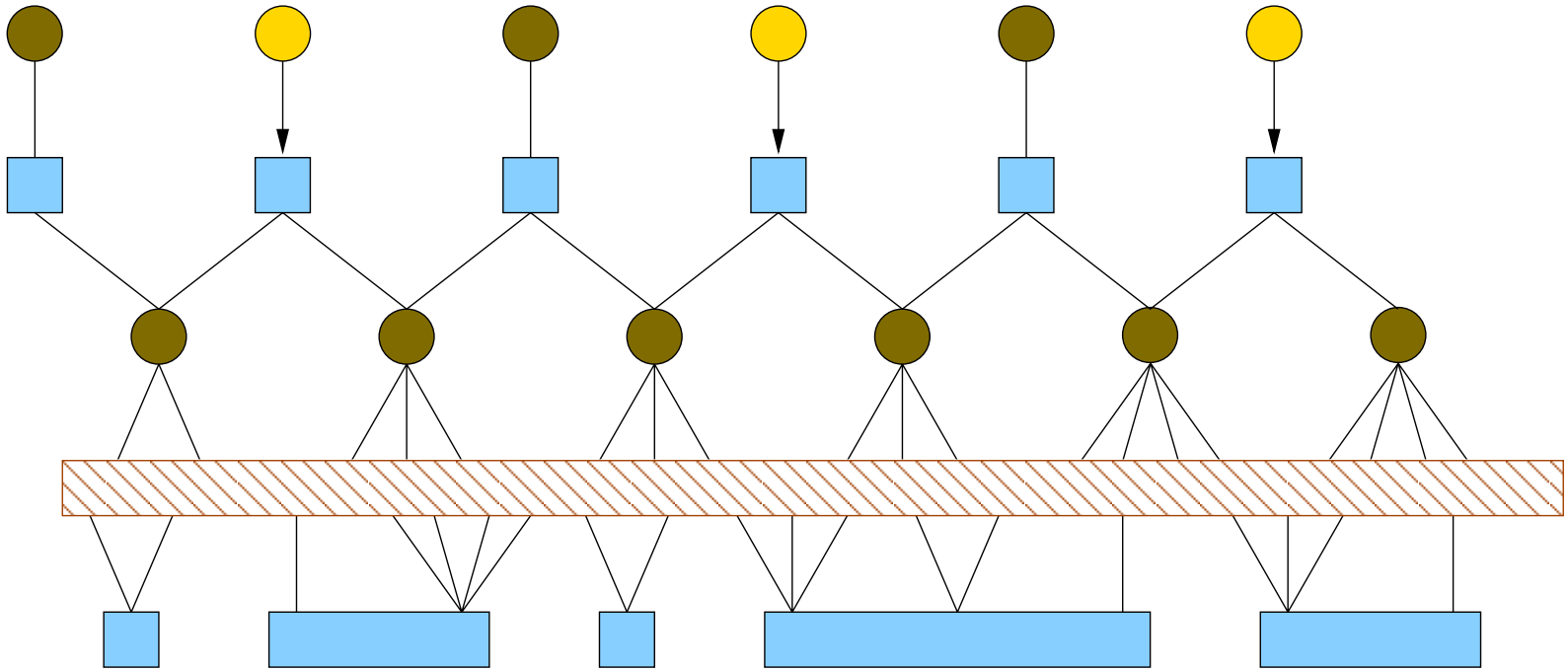
Sum check equations to remove

Example of Graph Reduction



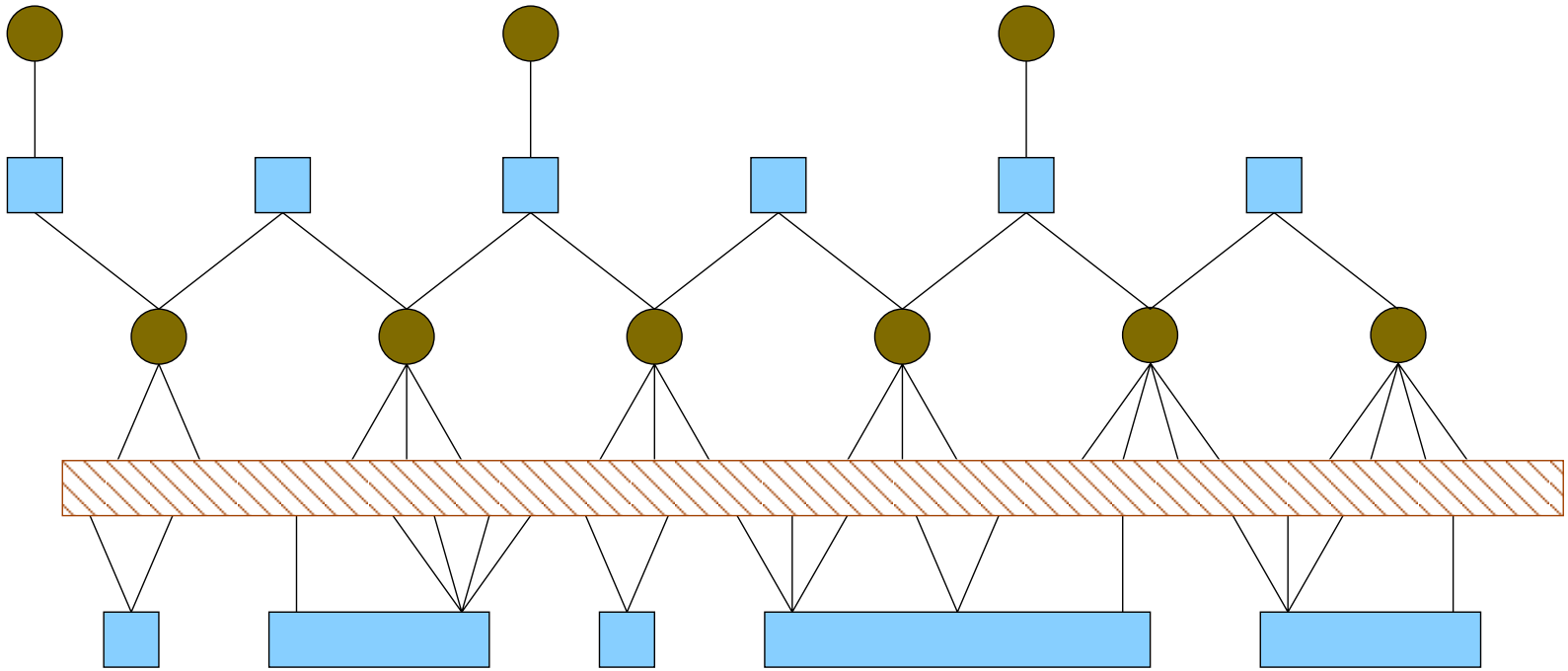
Tanner graph after check node graph reduction

Example of Graph Reduction



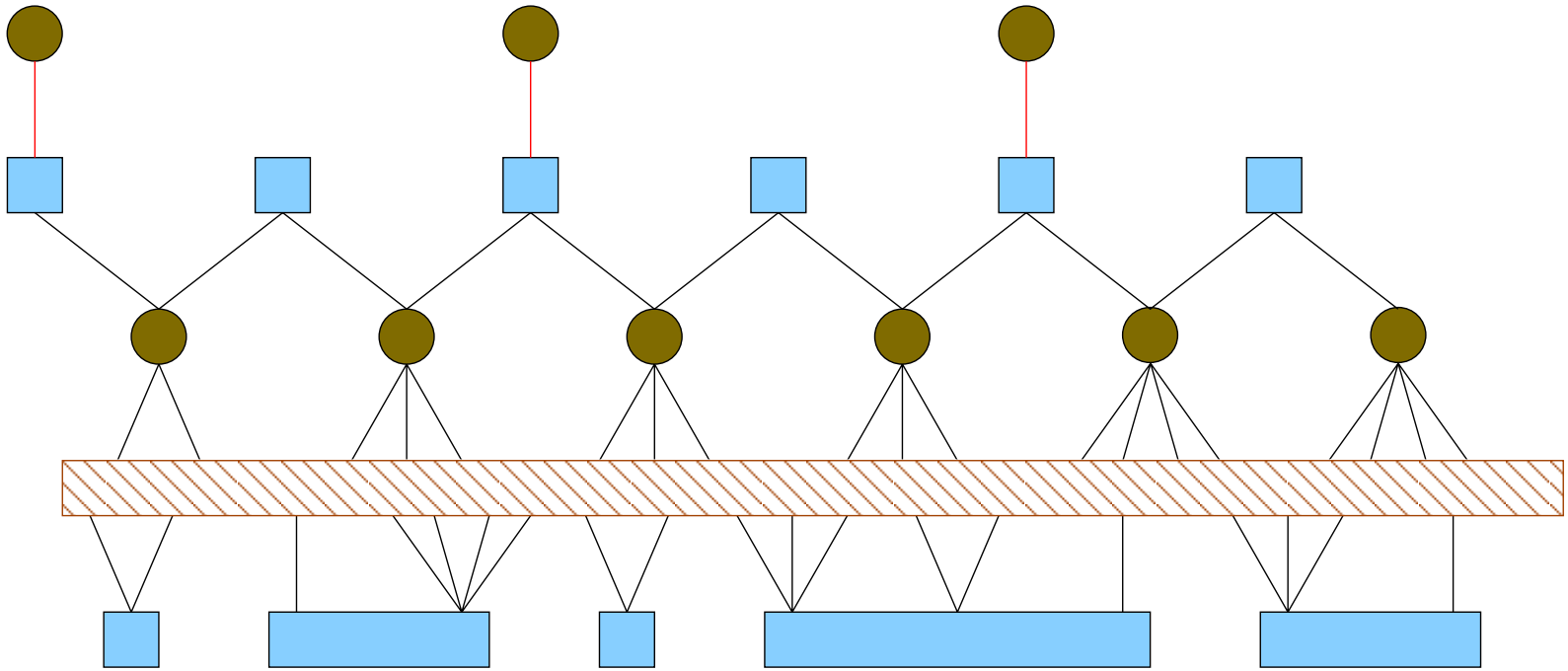
Mark known systematic bits

Example of Graph Reduction



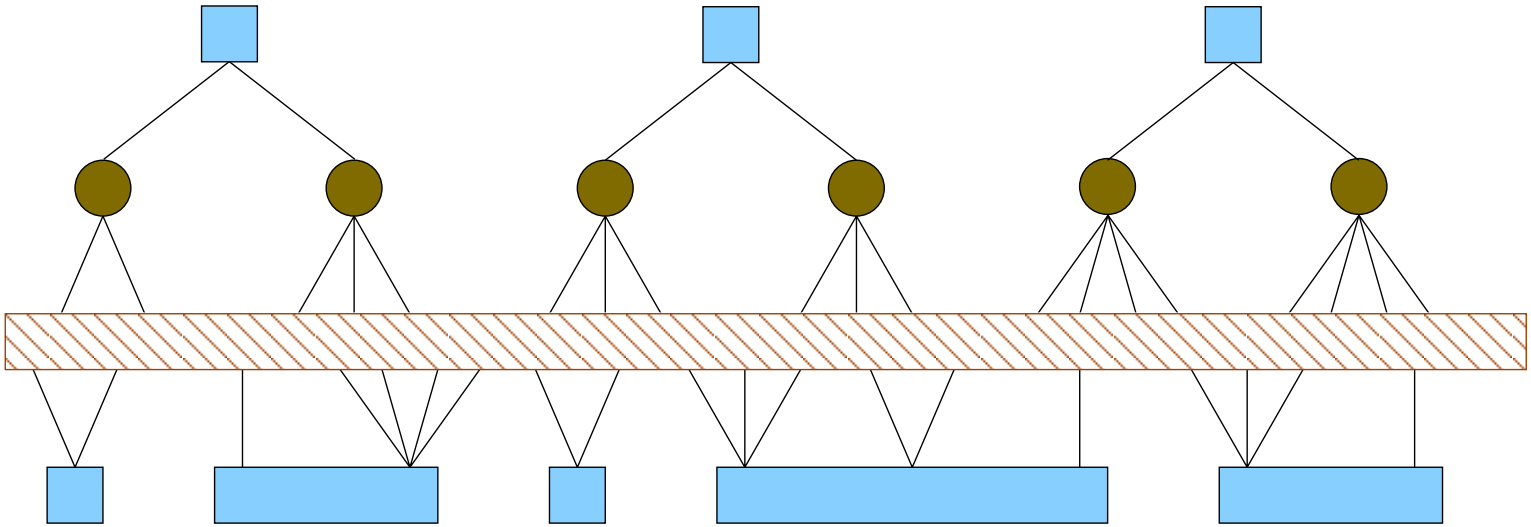
Merge values into checks

Example of Graph Reduction



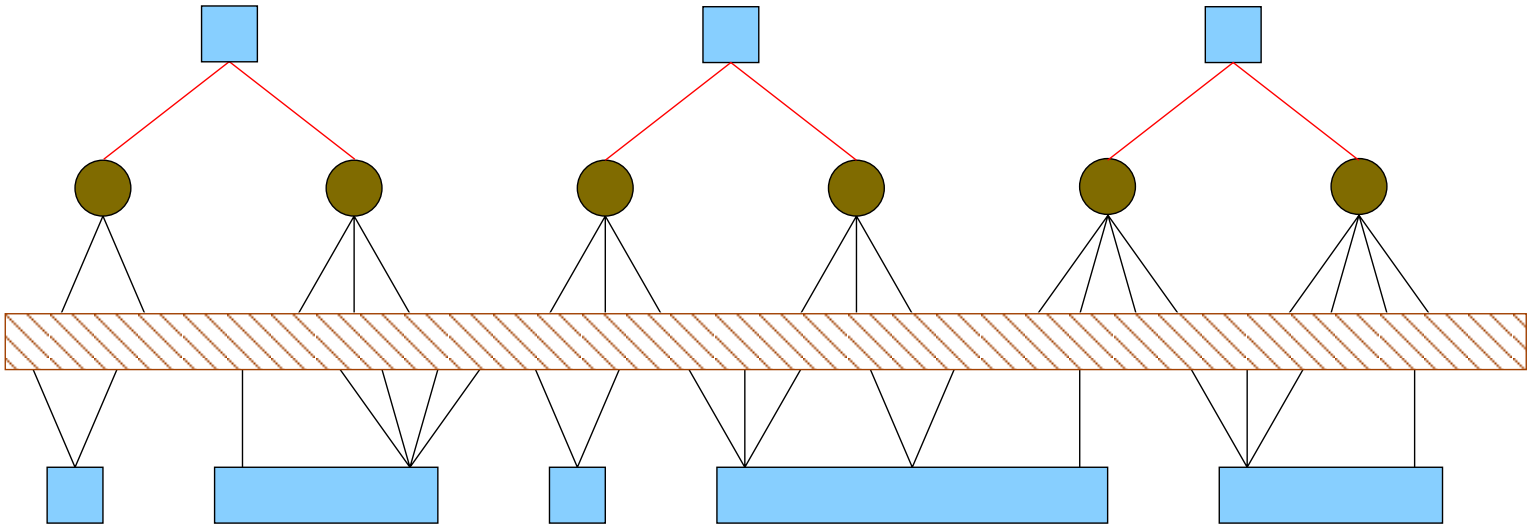
Mark unknown systematic bits

Example of Graph Reduction



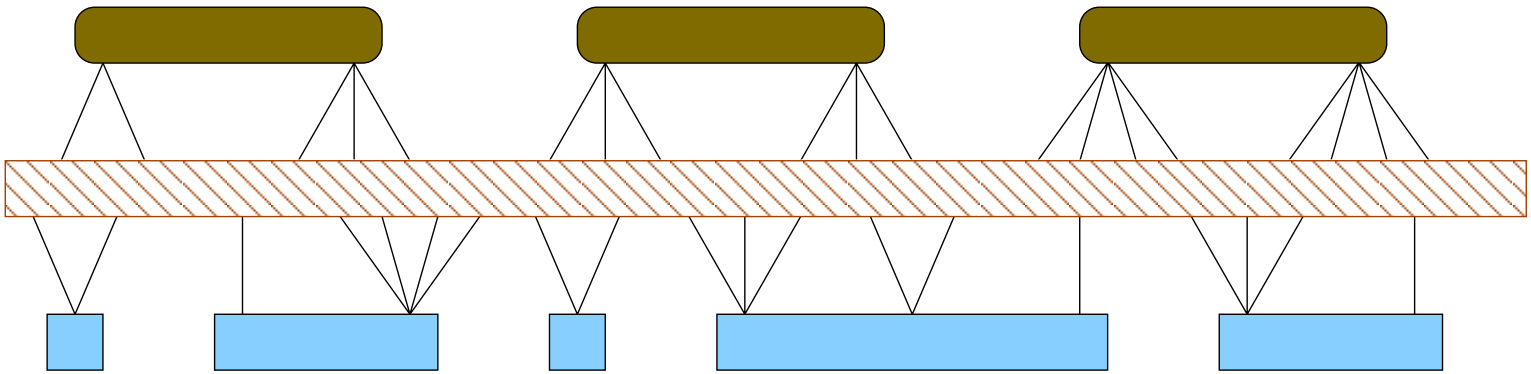
Remove unknown systematic bits

Example of Graph Reduction



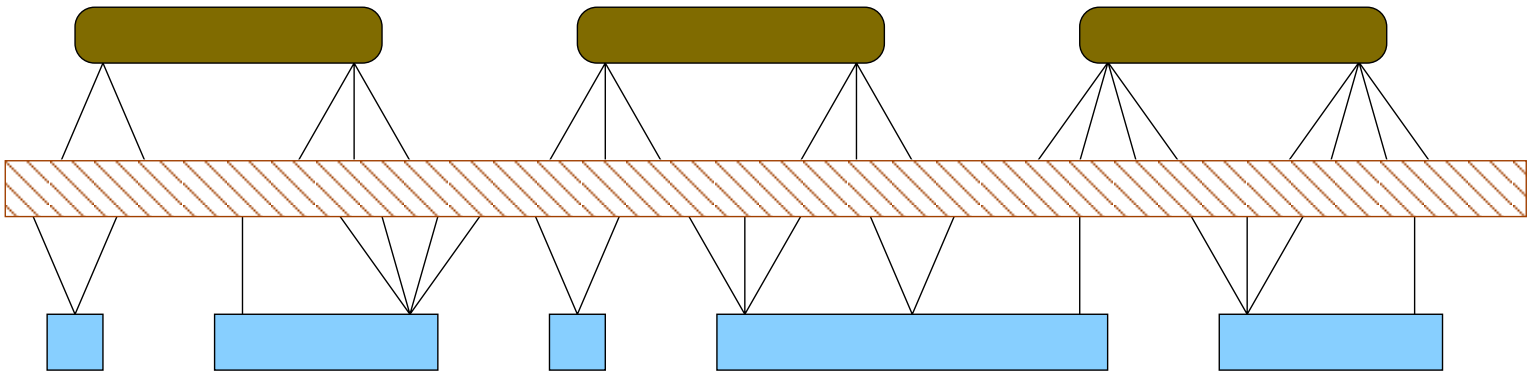
Mark degree 2 check nodes

Example of Graph Reduction



Combine bit nodes to remove

Example of Graph Reduction



Tanner graph of residual LDPC

Density Evolution via Graph Reduction for ARA Codes

After the graph reduction, we are left with a standard LDPC ensemble whose new edge-perspective degree distributions are given by

$$\widetilde{\rho}(x) = \frac{\widetilde{R}'(x)}{\widetilde{R}'(1)} = \frac{(1-p)^2\rho(x)}{\big(1-pR(x)\big)^2}$$

$$\widetilde{\lambda}(x) = \frac{\widetilde{L}'(x)}{\widetilde{L}'(1)} = \frac{p^2\lambda(x)}{\big(1-(1-p)L(x)\big)^2}$$

- Swapping p with $1-p$ exposes a nice symmetry between the information and parity bits

Capacity-Achieving ARA Codes (1)

- Suppose we choose the d.d. after graph reduction to be

$$\widetilde{\lambda}(x) = \widetilde{\rho}(x) = \frac{(1-b)x}{1-bx} \qquad 0 < b < 1.$$

- Since $\widetilde{\lambda}\left(1-\widetilde{\rho}(1-x)\right) = x$, this choice gives a c.a. LDPC ensemble after graph reduction
- Inverting the graph reduction to get the original d.d. gives

$$L(x) = \frac{bx + \ln(1-bx)}{p\left[b + \ln(1-b)\right] + (1-p)\left[bx + \ln(1-bx)\right]}$$

$$R(x) = \frac{bx + \ln(1-bx)}{(1-p)\left[b + \ln(1-b)\right] + p\left[bx + \ln(1-bx)\right]}.$$

Capacity-Achieving ARA Codes (2)

Theorem (Self-Matched ARA Codes)

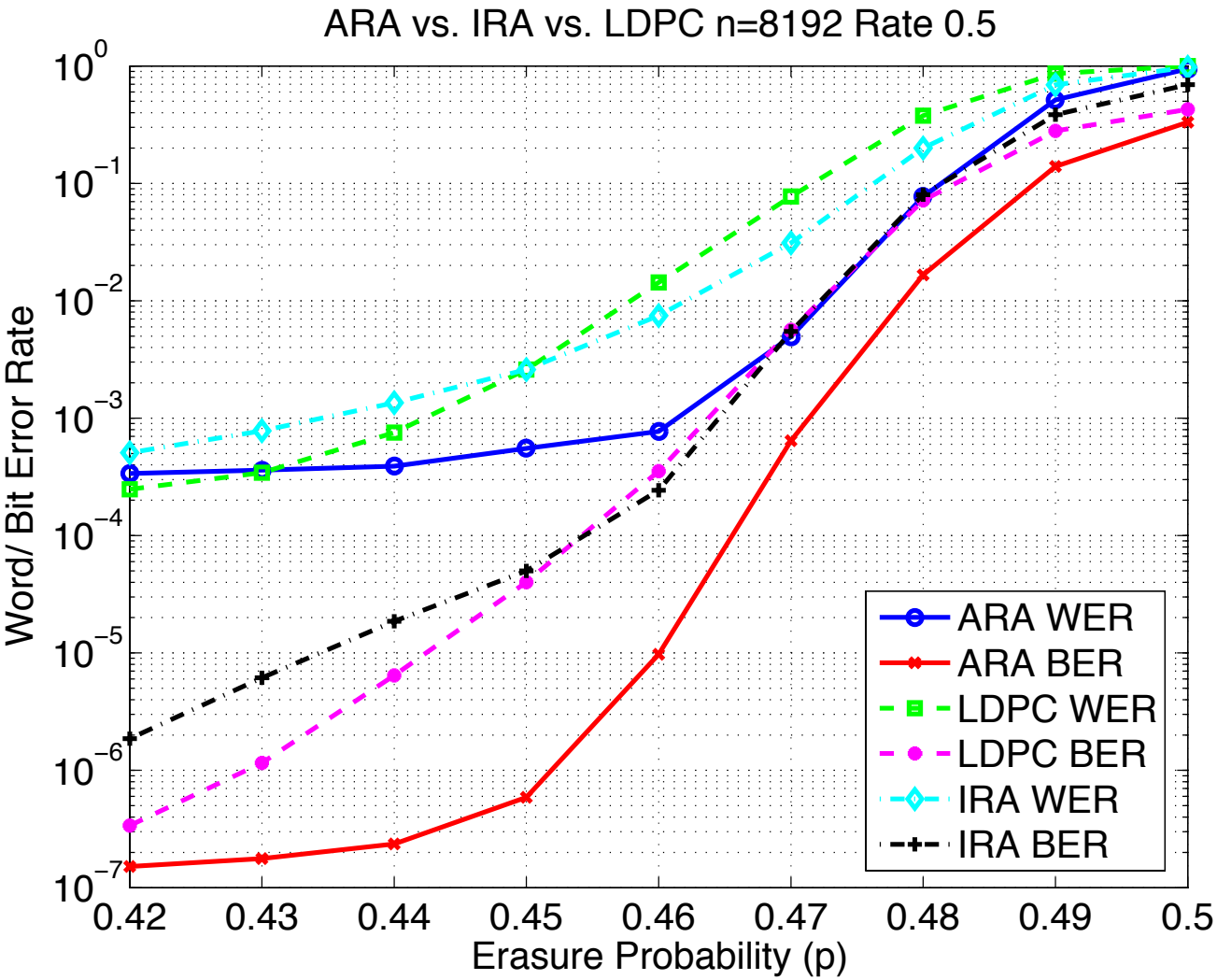
The power series expansions of $L(x)$ and $R(x)$ are non-negative for $p \in (0,1)$ if b is chosen (in terms of Lambert W-function) to be

$$b = W\left(-e^{-\frac{13+\sqrt{61}}{12}\frac{1+|1-2p|}{1-|1-2p|}-1}\right)+1.$$

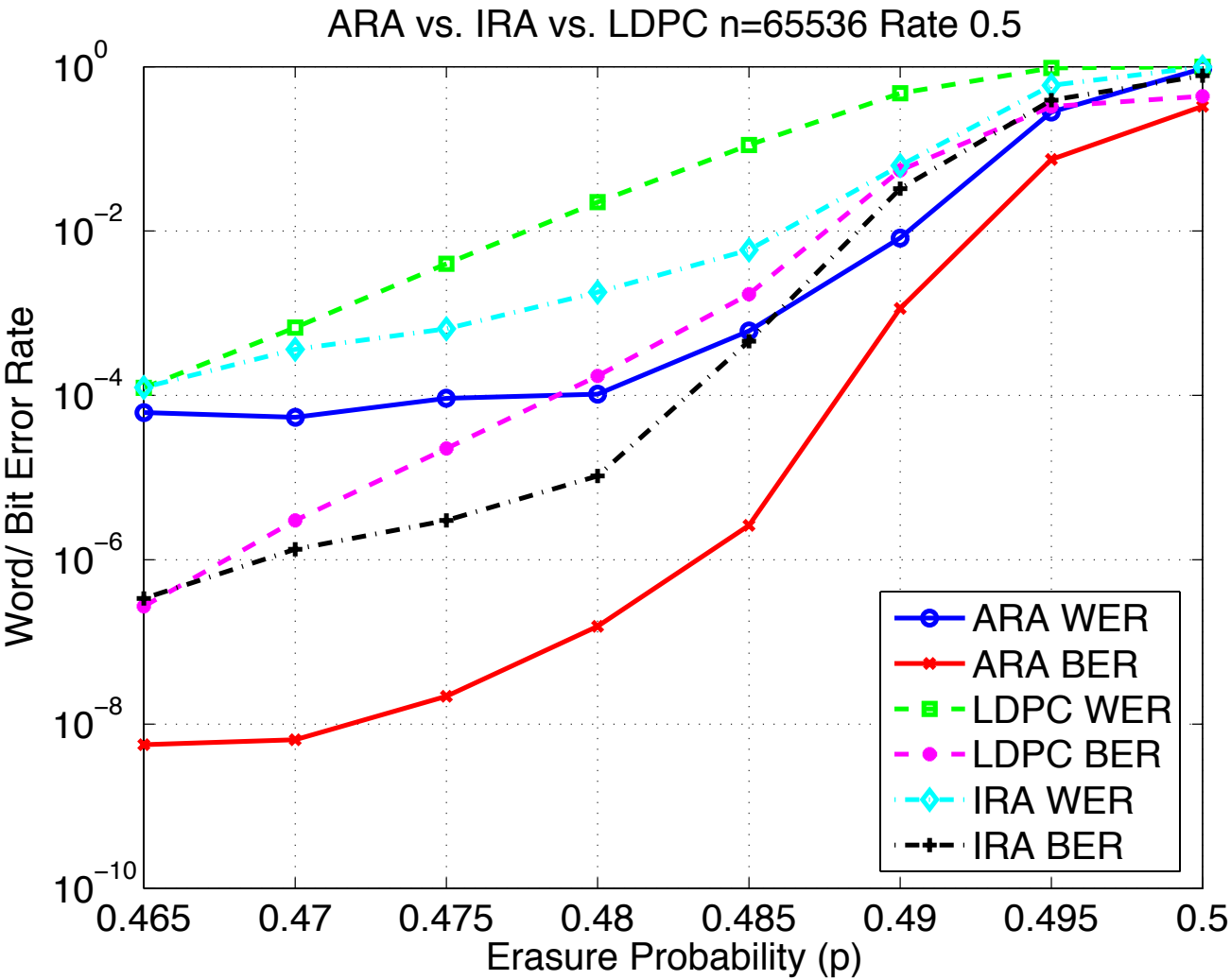
Therefore, the ARA ensemble defined by (L,R) achieves capacity on the BEC under iterative decoding for $p \in (0,1)$.

Moreover, the tails of the d.d. decay exponentially fast and the encoding/decoding complexity is bounded.

Computer Simulations (1)



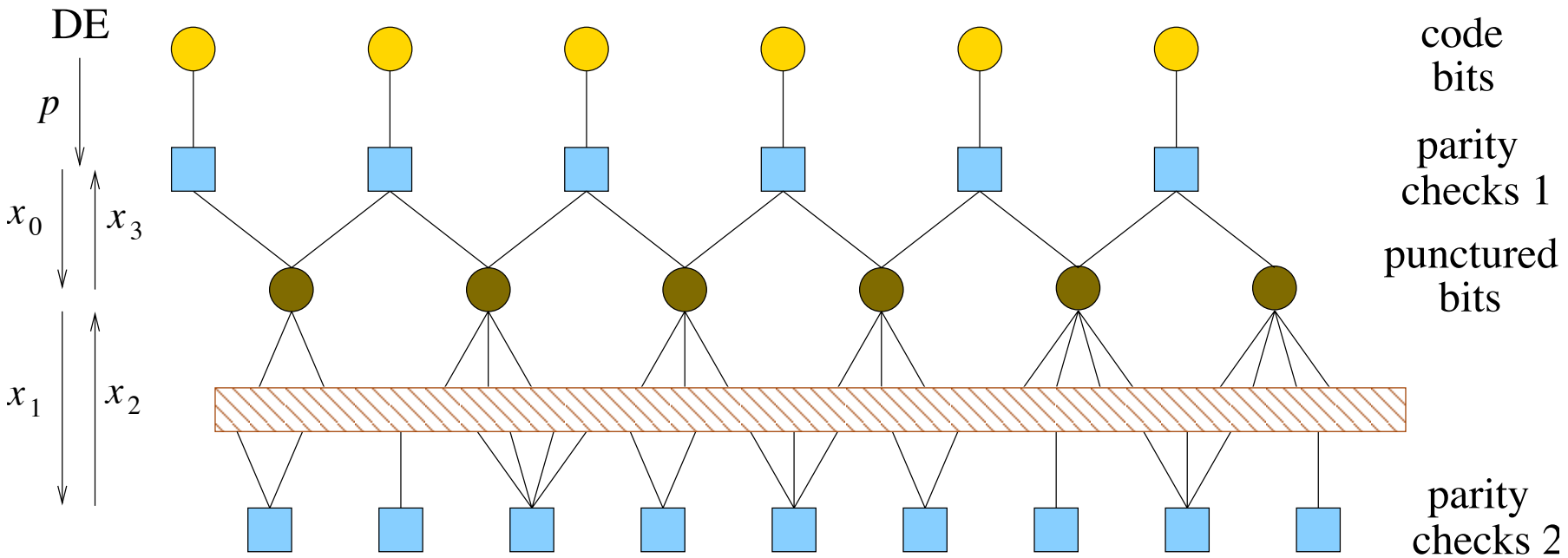
Computer Simulations (2)



Forney's Duality Transform

- General Transform for Forney Style Factor Graphs
  - Maps a graph for code $\mathcal{C}$ to a graph for dual code $\mathcal{C}_\perp$
- For LDPC-Type Codes (i.e., only bit and check constraints)
  - Simply swap bit and check nodes
  - Leave channel observations connected to original node
- Applied to LDPC-Type Ensembles
  - Swap bit and check degree distributions
  - Maps c.a. ensembles for p to c.a. ensembles for $1-p$
  - Can also be seen as a symmetry of the DE equations
- Examples
  - LDPC ensembles map to low-density generator matrix ensembles
  - ARA ensembles map to ARA ensembles
  - NSIRA ensembles map to ALDPC ensembles

Accumulate LDPC (ALDPC) Codes



- After "accumulate", code bit sequence belongs to an LDPC code
- Natural image of NSIRA codes under the duality transformation
- Graph reduction only on bit d.d. (versus check d.d. for NSIRA)

Capacity-Achieving ALDPC Codes

- Swap the Bit and Check Nodes of the NSIRA Ensemble
  - Accumulate graph moves from check nodes to bit nodes
  - Bit and check d.d. ($L(x)$ and $R(x)$ are swapped)
  - Symmetry of DE equations shows new ensembles is c.a. for $1-p$
- The Bit-Regular ALDPC Ensemble
  - Dual ensemble of the check-regular NSIRA ensemble
  - Encoder first encodes the LDPC code and then differences bits
  - Minimum bit-degree of 3 implies LDPC d_{min} grows linearly
  - Difference operation shouldn't affect the minimum distance

Summary

- Introduced Various C.A. Codes with Bounded Complexity
  - ARA Codes: Systematic codes with bounded complexity
  - ALDPC Codes: Good minimum distance and bounded complexity
  - Simulations show ARA superior to other c.a. ensembles
- Introduced Density Evolution Via Graph Reduction
  - Exposes natural symmetry between LDPC, ARA and NSIRA codes
  - Allows c.a. LDPC codes to be mapped onto other code structures
- Full paper: H. Pfister and I. Sason, “Capacity-achieving ensembles of accumulate-repeat-accumulate codes for the erasure channel with bounded complexity", submitted to *IEEE Trans. on Information Theory*, December 1st, 2005. [Online at the arXiv]