EECE 580B Modern Coding Theory
WWW: http://dde.binghamton.edu/filler/mct/hw/5

Homework assignment #5
Due date: October 29, 2:50pm

Encoding of LDPC codes

Encoding of LPDC codes are done in two parts. The goal of this assignment is to implement both parts in Matlab. You are asked to implement two Matlab functions ("ldpc_preprocess.m" and "ldpc_encode.m"). First function will take care of the preprocessing step and store ALL necessary results to a Matlab structure (see below). This structure is then passed to the second function along with the vector of information bits. We will use the following structure to store all necessary results:

```
code_struct.H          % original parity-check matrix H
code_struct.row_perm   % row permutation and
code_struct.col_perm   % column permutation
code_struct.A          % matrix A (this notation is taken from the lecture)
code_struct.B          % matrix B (this notation is taken from the lecture)
code_struct.C          % matrix C (this notation is taken from the lecture)
code_struct.D          % matrix D (this notation is taken from the lecture)
code_struct.E          % matrix E (this notation is taken from the lecture)
code_struct.T          % lower triangular matrix T
code_struct.g          % gap = height of matrix D (or C or E)
code_struct.F          % matrix F = D-E*T\B % make sure this is a full rank matrix
code_struct.F_inv      % inverse of matrix F
```

Problem 1 (Preprocessing step):
Let C be an LDPC code described by a sparse parity-check matrix H. Implement a Matlab function "ldpc_preprocess.m" that, for given H, finds row and column permutation such that the matrix H is transformed into the following form

$$\widehat{H} = \begin{pmatrix} A & B & T \\ C & D & E \end{pmatrix},$$

where T is a lower triangular matrix with ones on the diagonal and the matrix $F = ET^{-1}B$ is of a full rank. The matrices are of the following size: $A \in GF(2)^{n-k-g \times k}$, $B \in GF(2)^{n-k-g \times g}$, $C \in GF(2)^{g \times k}$, $D \in GF(2)^{g \times g}$, $E \in GF(2)^{g \times n-k-g}$, $T \in GF(2)^{n-k-g \times n-k-g}$. Calculate F^{-1}.

I expect the function in the following form:

```
function [ code_struct ] = ldpc_preprocess( H )
% code_struct is the Matlab structure as described above.

  code_struct.H = H;     % store the matrix to the structure
  code_struct.row_perm   % row permutation and
  code_struct.col_perm   % column permutation
end
```

Use matrices H1,…,H11 from "test_matrices.mat" (see web page) for testing before you submit your homework. This MAT file contains sample output structures code_struct1,…, code_struct11. Your algorithm may give you different permutations and matrices! This is just to see how the output structures may look like. You may use them in the second problem for testing.

As an output of this problem I want the following:

- Complete code listing of the function "ldpc_preprocess.m" and all other functions you use.
- MAT file "hw5_results1.mat" produced by the following test script:

```
clc; clear; load('matrices1.mat'); % download 'matrices1.mat' from the web site
code_struct1 = ldpc_preprocess( H1 );
code_struct2 = ldpc_preprocess( H2 );
code_struct3 = ldpc_preprocess( H3 );
save('hw5_results1.mat');
```

Hints, warnings and suggestions:

- Start with the matrix H1 from "test_matrices.mat" file. See

```
load('test_matrices.mat');
spy( H1(code_struct1.row_perm, code_struct1.col_perm ) )
```

- Use functions "binary_inv.m" and "binary_rank.m" to calculate inverse and rank of a matrix in binary arithmetic. Download these functions from the web site. These functions use a modification of the Gaussian elimination algorithm to calculate the required quantities.

```
X = [1 1 1; 0 1 0; 1 0 0]
binary_rank(X) % output = 3
X_inv = binary_inv(X)
mod(X*X_inv,2)
```

Problem 2: (Encoding of information bits)

Given the decomposition of the parity-check matrix H into matrices A, B, C, D, E, T, F^{-1}, implement the encoding algorithm described in Part 2 of your lecture notes. That is, given k-bit vector of information bits s, find a codeword x satisfying $Hx^T = 0$. I assume the Matlab function in the form:

```
function [ x ] = ldpc_encode( s, code_struct )
  % x = output codeword
  % s = vector of information bits
  % code_struct = structure carrying all the results from the preprocessing step
end
```

As an output of this problem I want the following:

- Complete code listing of the function "ldpc_encode.m" and all other functions you use.
- MAT file "hw5_results2.mat" produced by the following test script:

```
clc; clear; load('matrices2.mat'); % download 'matrices2.mat' from the web site
x1 = ldpc_encode(s1, code_struct1); sum1 = sum(mod(H1*x1',2));
x2 = ldpc_encode(s2, code_struct2); sum2 = sum(mod(H2*x2',2));
x3 = ldpc_encode(s3, code_struct3); sum3 = sum(mod(H3*x3',2));
save('hw5_results2.mat');
```

Hints, warnings and suggestions:

- Use the example given on the lecture first. Make sure $Hx^T = 0$.
- Make sure you permute x back using the inverse permutation as described in the notes.