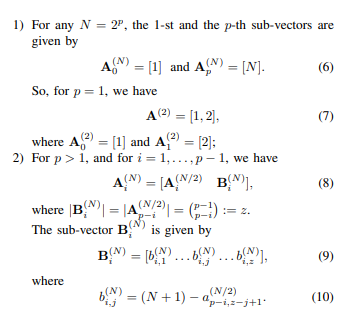
**Explaining the code**

**Part 1:reliaSeq (code used to generate the reliability sequence)**

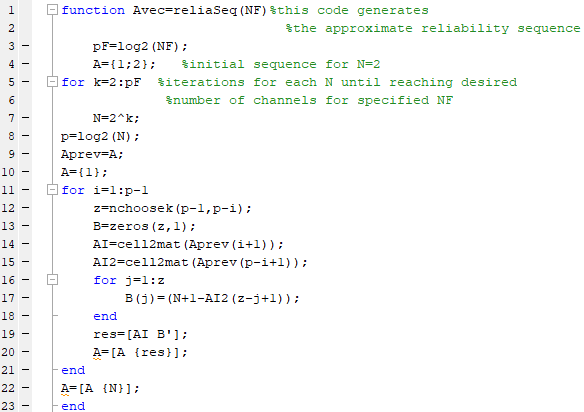
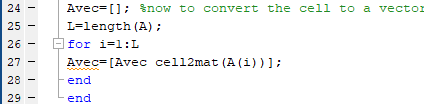
**Why?** The reliability sequence is a sequence of numbers that essentially sorts the channels by how well they perform at transmitting the data. It is sorted from worst to best channels. The leftmost (worst) channels will be frozen and the rightmost (best) channels will carry the message sequence

The reliability sequence must be obtained in order to perform polar decoding and encoding.

Normally, the reliability sequence is predefined by 5G standards.

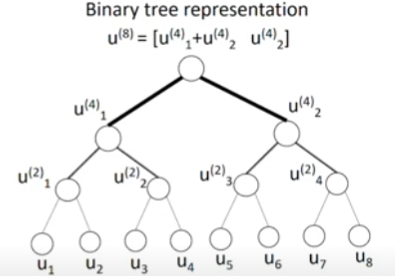


However, this was difficult to come by, and instead I found an approximate algorithm that claims it is 98% accurate (will be provided in the end).

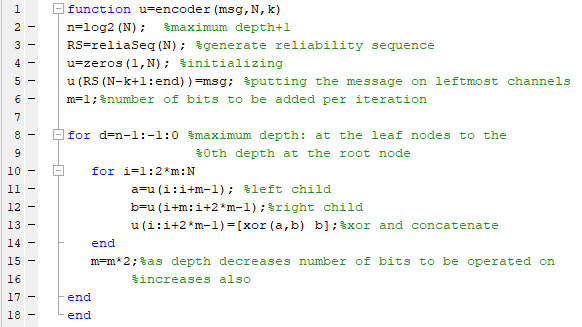
**Here is the matlab code implementation**

**Part 2: The encoder**

First, after determining the frozen channels and message carrying channels, we put the message in the rightmost channels. Note that for a message length K, K must not exceed the codeword length N. Note that N must be a power of 2.



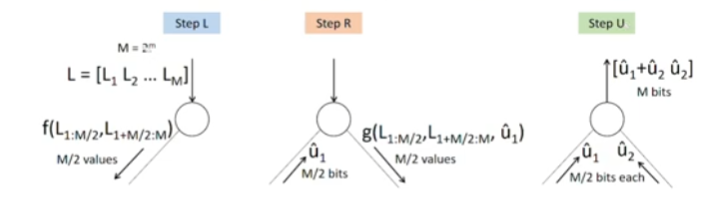
The algorithm is very simple. If we think of the codeword (u) as leaves in a binary tree, all that is needed to do is successive XOR and concatenation operations. For example, to get the bits at a certain nonleaf node, we XOR both of its children and concatenate the right child. This is done until the entirety of the codeword goes through this process. The maximum depth is determined by log2(N). Note that the number of bits XOR’d this way doubles every iteration.

**This is the matlab code**

**Part 3: The decoder by Successive Cancellation**

This one was tricky.

The decoder will be carried out in 3 main steps



**Step L “left”:** the interior node is given an array of M incoming “beliefs”. When direction to the left child the “min sum” function (f),specified as follows:

is applied on the two halves of the belief vector. This yields M/2 decisions on u1 to be sent back to the interior node and to the right child.

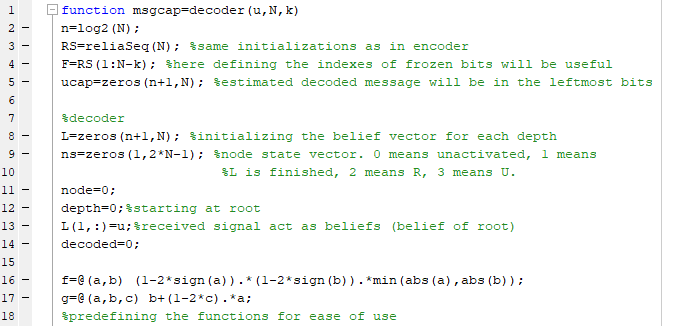
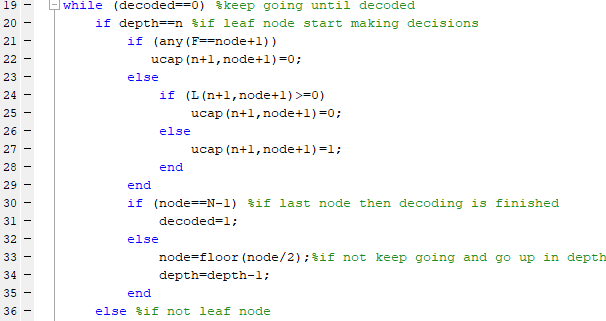
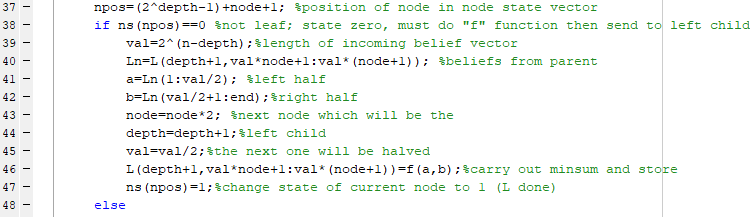
**Step R ”right”:** The interior node sends the same M beliefs to the right child in order to compute the “g” function specified as follows

Where r1 and r2 are as before, the two halves, and b is the left child decisions computed in the previous step. This yields another M/2 vector of decisions of the right node.

**Step U”up”:** When the interior node gets both of the decision vectors, it carries out the XOR then concatenate function on them, the same one used in encoding. This will yield an M length vector. Then it sends it up back to the parent. This M length vector will act as the decision estimate for the parent interior node (either u1 or u2 depending on if its left or right child), and this continues iteratively until the root depth is reached and the entire codeword is decoded, and a message is obtained

**In summary, the decoder works like this;**

* Start at root node.
* If leaf node, make decision and go to parent.
* If not leaf node, do step L and go to left child.
* Wait until decision is received from left child. Start step R when done
* Wait until decision is received from right child. Start step U when done
* Go to parent
* Back to beginning until done

MATLAB implementation will be as follows

