CSC8501 Coursework 2 – 2016
Convolutional Encoding
Due 28th October 2016 at 10am

Specification (what you need to do): You will build a computer program in C++ that decodes all the different encoded output files from coursework 1. You will demonstrate your program in the presence of error free and erroneous input data.

Design constraints:

- Use the Viterbi approach in your decoding solutions.
- For each encoded input file produce two decoded outputs in two separate text files: one without error injection and one with error injection.
- Error injection should be achieved using the following algorithm and should be carried out on the encoded input file before it enters the decoder:

    - BURST_MODE = FALSE
    - Burst_Mode_Counter = 0
    - WHILE !EOF
        - If Burst_Mode is TRUE then
            - Input_Bit becomes 1 or 0 (chosen randomly)
            - increment by 1 the Burst_Mode_Counter
            - if Burst_Mode_Counter > 4 then
                - Burst_Mode is FALSE
                - Burst_Mode_Counter is 0
            - endif
        - elseif
            - Random_Var is any number between 0 and 20
            - If Random_Var is greater than 18 then
                - BURST_MODE is TRUE
            - Endif
        - Endif
    - REPEAT

Deliverables (what we want to see):

- C++ source code authored by the student (*submit to submission system*)
- Executable file containing solution (*submit to submission system*)
- Output files produced by decoders and their comparisons to the original input file (*for demonstration only*)

Demonstration (your chance to explain and show your solution):
On Friday 28th October from 10am onwards students will demonstrate their solutions.

Learning Outcomes (what we expect you to demonstrate in a general way)

- Be capable of designing and creating programs.
- Realise inappropriate/appropriate usage of programming languages.
- Understand how to manage memory.
- To be able to create and use data structures.
- To be able to use condition statements, loops and functions.
- Identify appropriate techniques for analysing the efficiency of programs.
- To be able to utilise concurrency when appropriate.
- To be able to create programs that handle run-time errors.
- To be able to use appropriate techniques for debugging and analysing.
- To be able to design programs using a well-known methodology.

Marking Scheme (what is worth what):

Marks are out of 25 and are awarded as follows:

- 10 Marks for achieving correct output
- 5 Marks for appropriate file comparisons
- 5 Marks for user advanced features (parallel execution/templates)
- 5 Marks for adherence to the 7 rules of programming (see lecture 1)

Additional direction (making the task clearer):

- **Task 1**: Construct a computer program that will decode ALL the encoded files from coursework 1.
    - **You will need a solution per-encoding!**
- **Task 2**: Construct a computer program that injects erroneous states into the encoded data contained in the files output in coursework 1.
    - **This is best done independently of decoding**. Run the algorithm over each data element in turn inside the encoded text file. This creates a suitable erroneous input file for your decoder. **Remember to keep an error-free version for the error-free input**!
- **Task 3**: Run erroneous and error-free encoded files through your decoder.
- **Task 4**: Construct a computer program that compares the two decoded files with the original input file (the one used to go into the encoders). This program should highlight if successful decoding took place and the degree that it didn't (if there are differences).
- **Task 5**: Viterbi lends itself to parallel execution so try to exploit this with a threaded approach to a solution.
    - **Get a non-parallel solution working correctly before tackling a parallel solution.**

Set by: Graham.Morgan@ncl.ac.uk