



TEORIA DA INFORMAÇÃO
Ano Lectivo 2019/2020

Sistema de Comunicação com Canal Ruidoso

Ana Silvério nº 37561
Miguel Luís nº 37555

1 Descrição

No âmbito da avaliação prática da disciplina, pretende-se a construção de um sistema de comunicação para enviar mensagens por um canal de comunicação ruidoso. Quer isto dizer que, o canal de comunicação irá corromper os bytes enviados pela fonte, podendo ou não trocar os bits.

O programa fonte simula uma fonte de informação e gera mensagens para o standard output. O mesmo ocorre com o canal, uma vez que existe um programa chamado canal que recebe bytes do standard input e envia os mesmos bytes, possivelmente corrompidos, para o standard output.

O trabalho consiste na criação de um modelo probabilístico, tanto para a fonte como para o canal, e obter um codificador e decodificador do canal recorrendo a um algoritmo de compressão dado nas aulas. Na realização do trabalho também se encontra englobada a criação de um método de deteção de corrupção que possa ocorrer pelo canal, uma vez que este, tal como foi mencionado, é um canal com ruído.

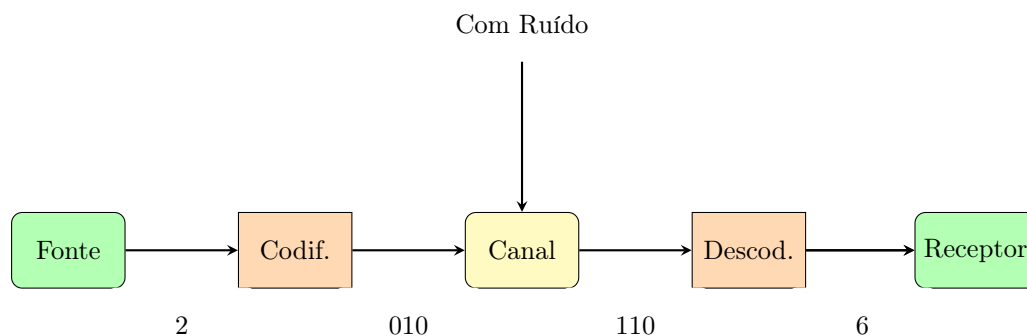


Figura 0: Canal com Ruído

2 Algoritmo Escolhido

Para a realização da codificação e decodificação de dados a implementar recorreu-se ao algoritmo LZW, uma vez que este algoritmo de compressão de dados ocorre sem perdas dos mesmos. Ao contrário de outros algoritmos, por exemplo o código de Huffman, o código LZW não necessita de um conhecimento á priori das probabilidades de ocorrência dos símbolos a comprimir.

O algoritmo LZW é um derivado do Algoritmo de Huffman juntamente com o Algoritmo LZ78. Logo, o que este algoritmo necessita para ser colocado em prática é a frequência com que os símbolos a codificar se repetem e não a frequência de ocorrências destes.

Dado a implementação deste algoritmo ser relativamente simples e compatível com todo o tipo de sistemas, esta foi mais uma das razões pelo qual se optou pela sua selecção.

3 Compressão

A compressão dos dados é feita a partir dos dados do programa Fonte e contém as seguintes funções:

1. **Compress:** Esta função recebe os dados do programa Fonte como argumento e vai comprimi-los para binário recorrendo ao algoritmo LZW.
2. **Write file:** Esta função vai receber como argumento os dados comprimidos pela função anterior e vai escreve-la num ficheiro de extensão .txt.

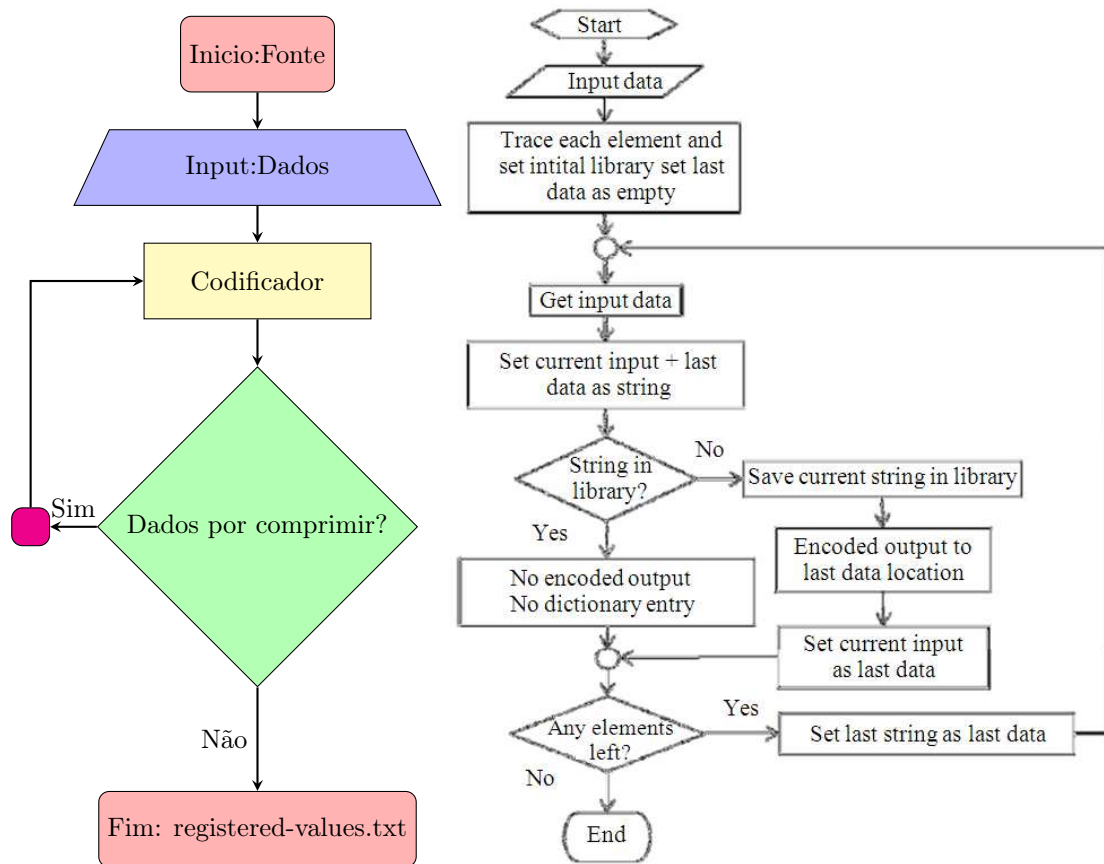


Figura 1: Flowchart - Metodologia de Processamento

Figura 2: Flowchart - Codificador Utilizado

4 Descompressão

A descompressão dos dados é feita a partir dos dados do programa Canal e contém as seguintes funções:

1. Organize Data: Esta função, tal como o nome indica, vai organizar os dados do programa Canal preparando-os para serem decodificados pelo decodificador. O seu argumento será os dados do programa.

2. Decompress: Esta função recebe como argumento os dados previamente comprimidos e vai fazer a acção contrária, vai descomprimi-los de acordo com o descompressor do algoritmo de LZW.

3. Hamming Correction: Esta função vai receber como argumento os dados descomprimidos pela função anterior e o resultante do ficheiro de código hamming-code.py e vai averiguar se a informação foi deturpada pelo canal. Quer isto dizer, se ouve algum bit que tenha sido trocado. Caso tenha ocorrido, então a função vai repor o bit(s) trocados na posição ou ordem correta. A aplicação e implementação do código de Hamming encontra-se feita no ficheiro de código referido.

4. Write file: Esta função vai receber como argumento os dados comprimidos pela função anterior e vai escreve-la num ficheiro de extensão .txt.

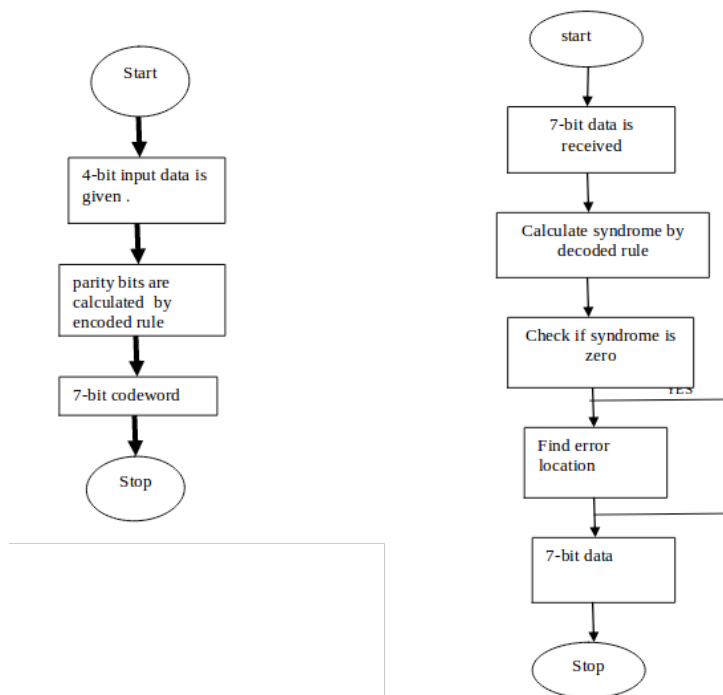


Figura 3: Flowchart - Processamento de um código de hamming

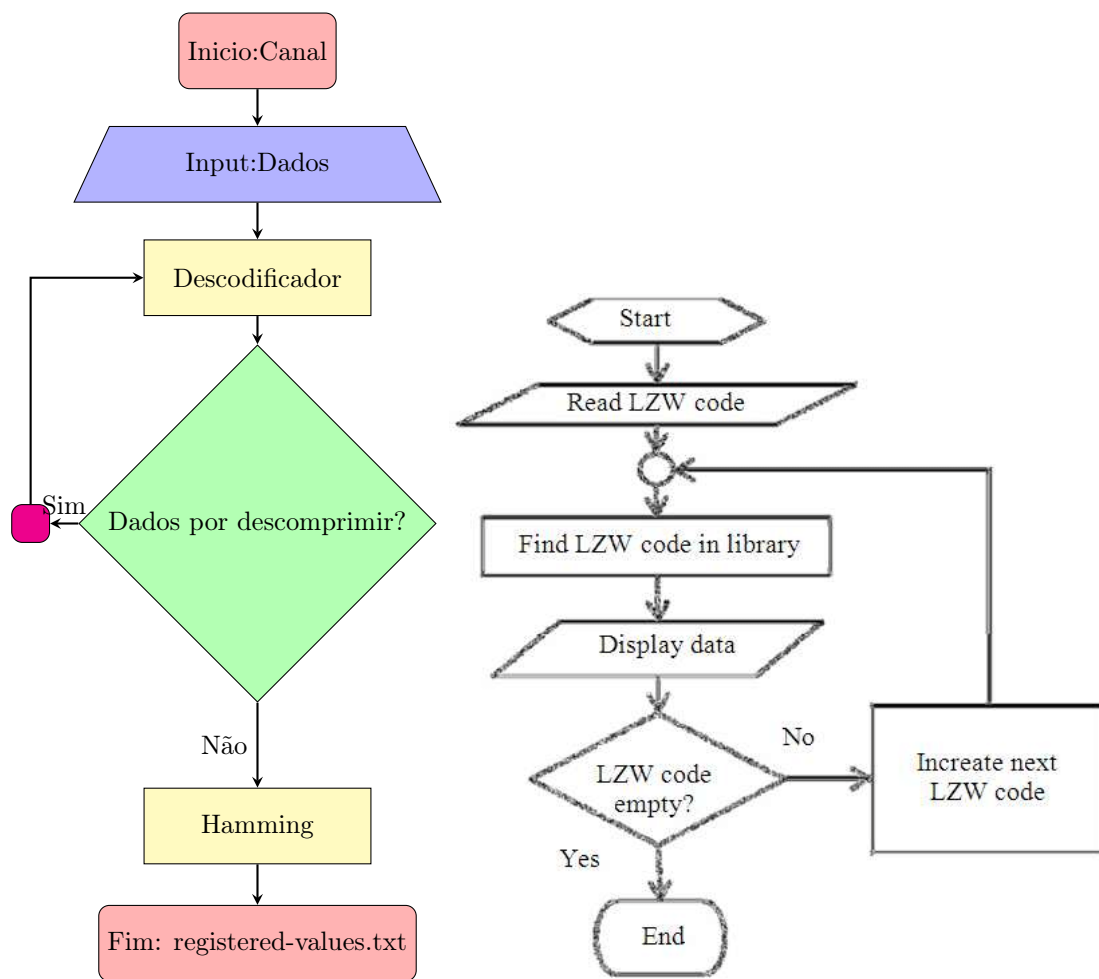


Figura 4: Flowchart - Metodologia de Processamento

Figura 5: Flowchart - Decodificador Utilizado

5 Cálculos

5.1 Entropia:

Formula Utilizada:

$$H(X) = - \sum_{x \in X} p(x) * \log_2 p(x)$$

5.2 Entropia Condicionada H(Y|X):

Formulas Utilizadas:

$$H(X, Y) = - \sum_{x \in X, y \in Y} p(x, y) * \log_2 p(x, y)$$

$$H(Y|X) = H(X, Y) - H(X)$$

5.3 Capacidade do Canal:

O canal vai ser com perdas logo, a sua capacidade é calculada da seguinte forma:

Formula Utilizada:

$$C = 1 - P_e$$

Onde P_e é a probabilidade de inverter um bit e C a capacidade do canal.

6 Balanço

Após a realização deste trabalho, adquirirmos um conhecimento mais consolidado acerca da matéria leccionada ao longo do semestre. Conhecimento este que passa pela implementação em código python do algoritmo seleccionado, algoritmo LZW, sob a forma de codificador e decodificador de dados provenientes da fonte e do canal, respectivamente.

Como o canal é um canal com ruído, quando se obteve a informação decodificada através do algoritmo de LZW teve-se que recorrer a uma implementação adicional do código de hamming de modo a remover e restaurar, se necessário, qualquer troca de bits que possa ter ocorrido.

Com os dados recolhidos pelo codificador e decodificador, será possível proceder ao calculo da entropia e entropia condicionada colocadas em prática através de uma implementação em código python. Recorrendo sempre aos conteúdos leccionadas nas aulas.