



Universidade Federal da Paraíba
Centro de Informática

Introdução à Teoria da Informação

Gabriel Teixeira Patrício - Matrícula: 20170170889

Sarah Andrade Toscano de Carvalho - Matrícula: 20170022975

João Pessoa, 2021.



Universidade Federal da Paraíba
Centro de Informática

Projeto Final: Compressor e Descompressor de dados - LZW (parte I)

Relatório parcial sobre o software de compressão e descompressão de dados, cujo funcionamento baseia-se no algoritmo LZW. Tal projeto está sendo desenvolvido durante a realização da disciplina de Introdução à informação, ministrada pelo professor Derzu Omaia, do Centro de Informática da Universidade Federal da Paraíba.

João Pessoa, 2021.

Sumário

Sumário	3
Introdução	4
Desenvolvimento	4
Algoritmo do compressor	4
Algoritmo do descompressor	5
Análise e Resultados	7
Corpus 16 MB e Vídeo	7
Razão de Compressão por K	7
Tempo de Processamento por K	7
Índices codificados por K	8
Conclusão	8
Referências	9

Introdução

O Algoritmo **LZW (Lempel–Ziv–Welch)**, resulta do melhoramento efetuado por Terry Welch dos algoritmos LZ77 e LZ78, os quais foram desenvolvidos por Jacob ZIV e Abraham Lempel, respetivamente. Inicialmente, foi desenhado a pensar no número de bits que se poderia reduzir aquando do envio de ficheiros armazenados nos discos, mas tem vindo a ser usado em muitos outros contextos. Trata-se de um algoritmo de compressão de dados baseado num dicionário que produz códigos de comprimento fixo associados a sequências de símbolos de comprimento variável. Os códigos gerados durante o processo, tanto de codificação como de descodificação, correspondem a sequências de símbolos cada vez mais longas, que vão sendo adicionadas ao dicionário. O LZW é um modo de compressão sem perdas (CSP), o que significa que o fluxo de dados descomprimidos é exatamente igual ao fluxo de dados originais, não se perdendo qualquer informação. [2]

Em resumo, o algoritmo LZW é um algoritmo de codificação de palavras cuja idéia principal é ir construindo um dicionário de símbolos ou palavras conforme o texto ou a informação vai sendo processado pelo algoritmo.

Desenvolvimento

Para realizar a compressão iniciamos o dicionário LZW com um tamanho de 256 caracteres da tabela ASCII. Durante o processo de análise do texto, vamos adicionando no dicionário novas palavras a partir de combinações de pelo menos 2 caracteres encontrados no texto. A cada nova combinação encontrada, é criada uma nova entrada no dicionário que será utilizada posteriormente para novas codificações. Ao final do processo de codificação, teremos um dicionário de combinações de palavras que poderá ser utilizado para codificar o texto.

Algoritmo do compressor

A lógica do algoritmo de compressão pode ser descrita em duas etapas principais, uma quando o carácter analisado encontra-se indexado no dicionário e outra quando ele não está. Considerando a construção inicial do dicionário como etapa primordial, o funcionamento dessas situações é detalhado com o diagrama em blocos a seguir:

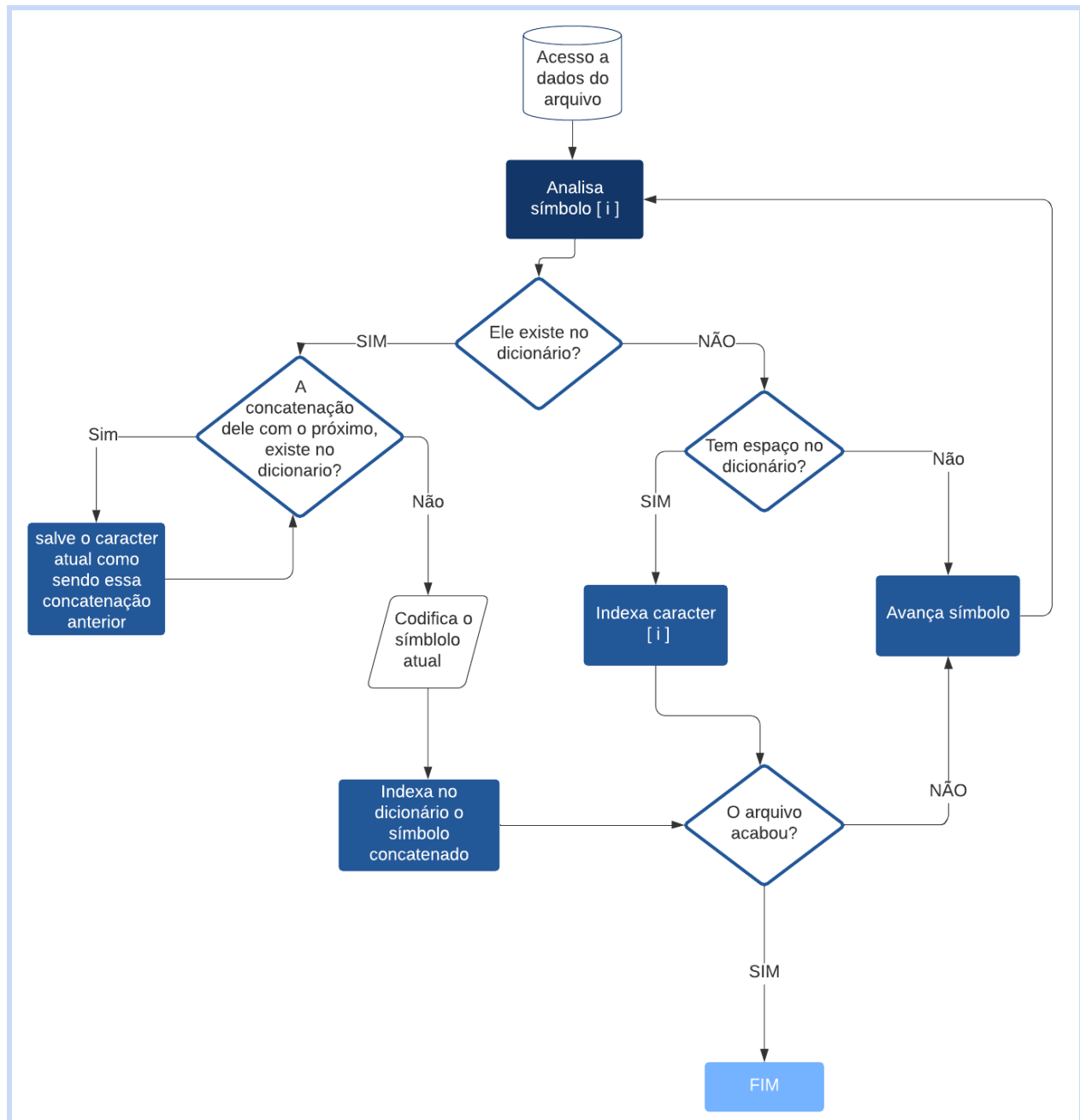


Figura 1. Diagrama em blocos do funcionamento do compressor pelo algoritmo do LZW.

Algoritmo do descompressor

O funcionamento do descompressor é similar ao do compressor, com algumas peculiaridades. Na leitura inicial do arquivo já é feita a decodificação do número binário para o seu inteiro equivalente e em seguida para o seu char equivalente na tabela ascii. Desse modo, vale ressaltar que no compressor, a modificação dos dados é feita apenas no final para escrevê-los no arquivo binário, enquanto no descompressor a primeira etapa já envolve tratamento nos dados recebidos.

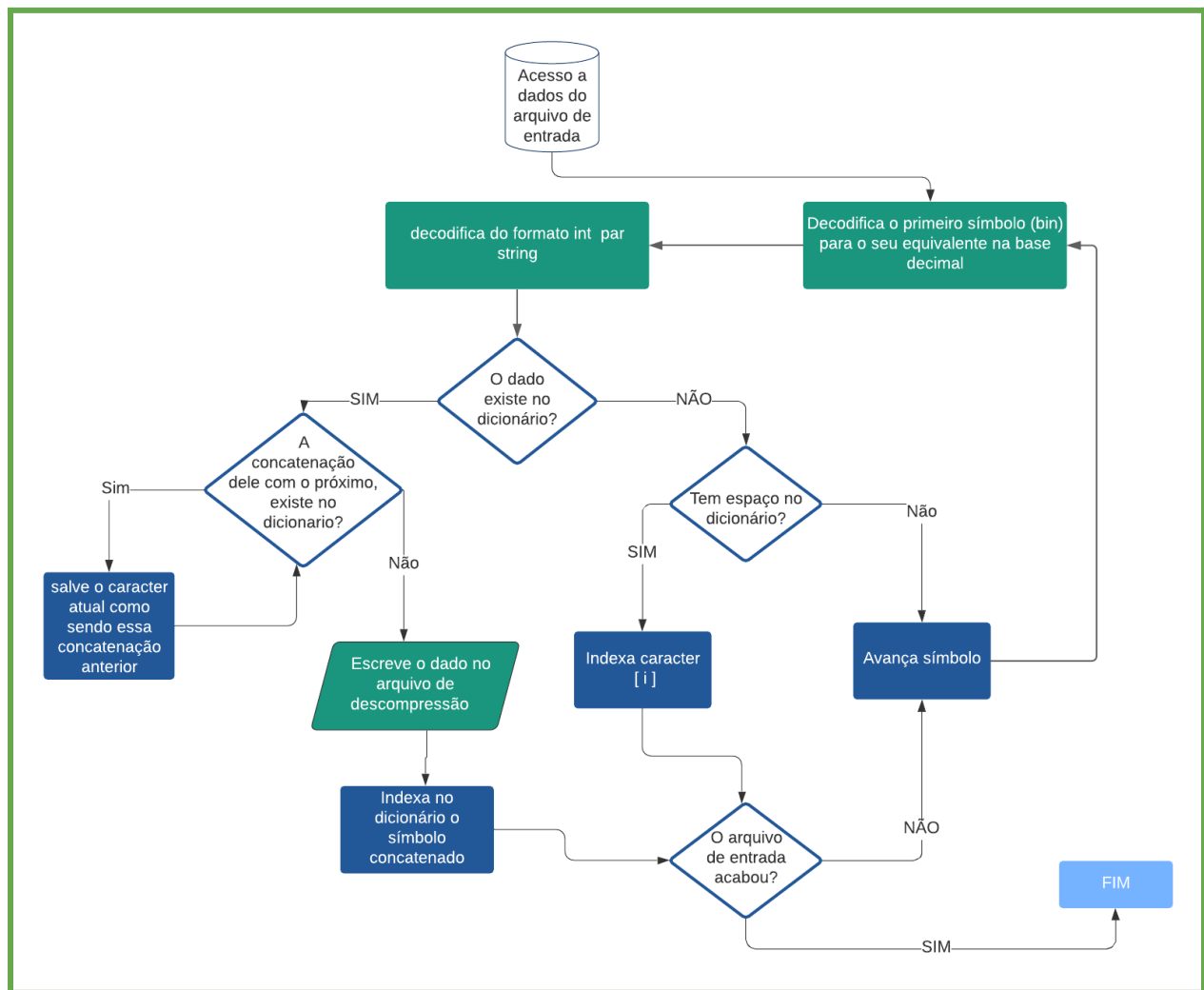


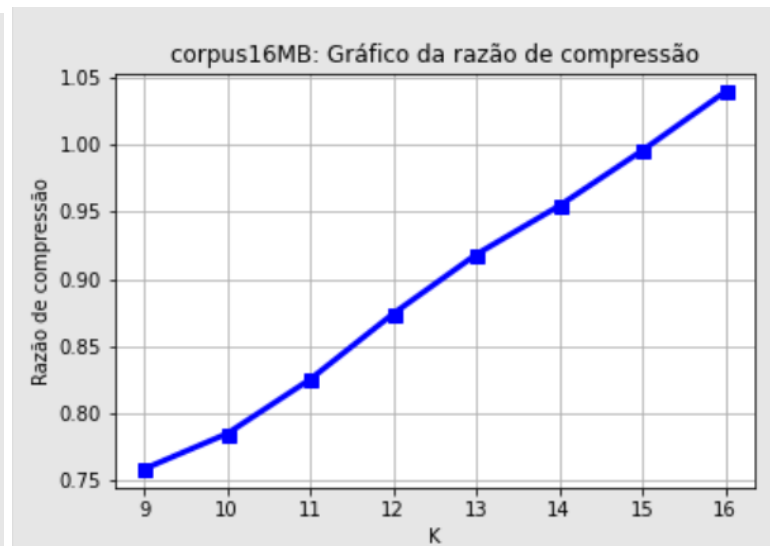
Figura 3. Diagrama em blocos do funcionamento do descompressor pelo algoritmo do LZW.

Análise e Resultados

Corpus 16 MB e Vídeo

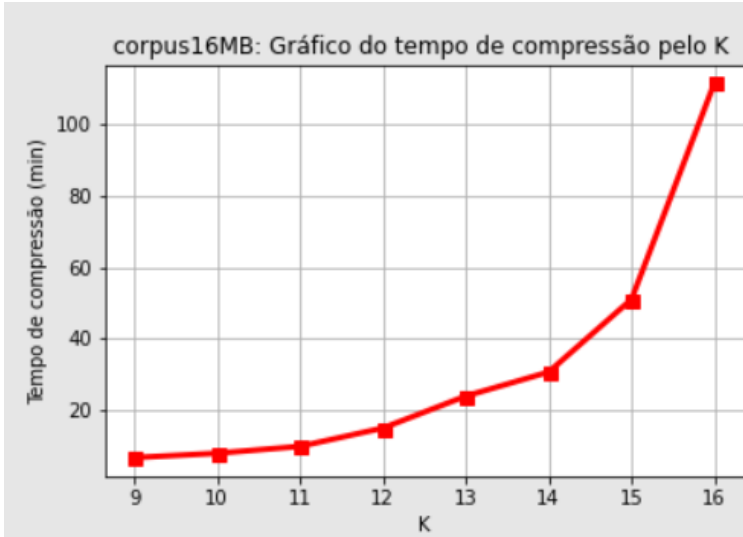
- Razão de Compressão por K

No gráfico a seguir podemos perceber que com o aumento do K a razão de compressão (que representa a quantidade de bytes do original referente ao comprimido) diminui para o caso do vídeo. Mas para o caso do texto essa razão é o contrário, quanto maior o K, maior a razão de compressão.



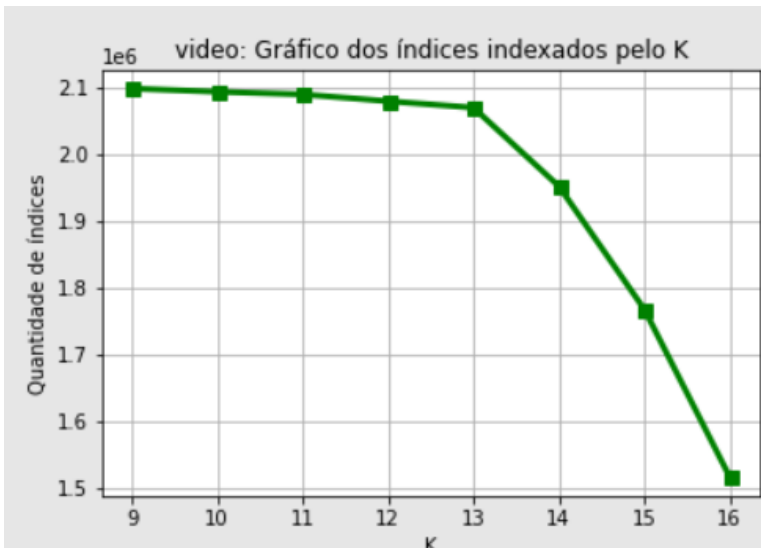
- Tempo de Processamento por K

Analisando os gráficos abaixo pode-se inferir que o compressor possui um tempo variavelmente linear para realizar a compressão dos dados. Nesse caso, as curvas são semelhantes, a diferença se evidencia na proporção do tempo. Como o vídeo é um arquivo menor, ele é comprimido mais rápido do que o arquivo do corpus.



- Índices codificados por K

Nos gráficos a seguir percebe-se que quanto maior o K, menor será a quantidade de índices indexados, pois o número disponível no dicionário é maior e não se tem a necessidade de uma quantidade maior de índices. Quando se tem um número pequeno de K, como por exemplo no caso da esquerda entre K9 - K13, o número do dicionário não foi suficiente para fazer a codificação e necessitando de mais índices.



Conclusão

Portanto, o compressor implementado nesta disciplina apresenta um tempo de compressão variável linearmente em relação ao número de Ks. Além disso, a quantidade de

índices codificados diminui quando o K aumenta, porém esse aumento pode ser leve ou mais brusco, dependendo das repetições de símbolos no arquivo comprimido.

Referências

1. <http://multimedia.ufp.pt/codecs/compressao-sem-perdas/codificacao-baseada-em-dicionarios/lzw/>
2. <https://www.ime.usp.br/~fli/gzip.html>
3. <http://www.ic.uff.br/~aconci/LZW.pdf>
4. https://www.decom.fee.unicamp.br/dspcom/EE088/Algoritmo_LZW.pdf
5. https://lucid.app/lucidchart/11576a09-73a4-4ada-b72d-8269be1d02c1/edit?viewport_loc=-205%2C746%2C2074%2C855%2C0_0&invitationId=inv_97c215ce-9790-46f8-9ba8-46b82b8dadf3
6. https://lucid.app/lucidchart/17c6c72d-ab2b-4143-b94a-a44540a4c726/edit?viewport_loc=-244%2C99%2C2414%2C996%2C0_0&invitationId=inv_22c30412-0d2e-4d30-93be-491d21e585c2