

OBTENÇÃO DOS CÓDIGOS CANÔNICOS

As propriedades dos códigos canônicos são: **a determinação dos comprimentos** dos códigos por meio do algoritmo de Huffman e que os códigos de mesmo comprimento são inteiros consecutivos.

A partir da obtenção dos comprimentos pelo algoritmo Moffat e Katajainen, para calcular o código basta ter em mente que **o primeiro código** é composto apenas por zeros e que para os demais, **adiciona 1** ao código anterior e realiza-se um **shift à esquerda** para que tenha o comprimento adequado quando necessário, de acordo com o vetor dos comprimentos dos códigos.

i	Símbolo	Código Canônico
1	rosa	0
2	uma	10
3	para	1100
4	cada	1101
5	, U	1110
6	É	1111

Tabela de códigos gerados

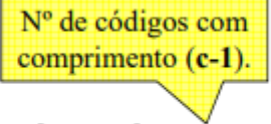
Para que eu possa realizar **a descompressão**, é necessário que eu armazene no arquivo comprimido uma tabela **semelhante** à tabela acima. O vocabulário é a primeira coisa a ser armazenada no arquivo comprimido. Como o vocabulário geralmente é grande, ele pode ocupar mais espaço do que o arquivo comprimido. **Para contornar esse problema**, há uma outra forma de chegar aos códigos guardando **apenas os vocabulários e posições das palavras** dentro do vocabulário e os códigos seriam gerados de forma dinâmica, apenas quando for necessário.

CODIFICAÇÃO E DECODIFICAÇÃO

Tendo como entrada, o resultado do algoritmo de Moffat e Katajainen, a próxima etapa é calcular dois vetores, Base e Offset, que são vetores indexados pelo comprimento do código, ou seja, se o maior código possui 4 bits, os vetores possuem 4 posições.

O vetor Base indica para um determinado comprimento, qual é o primeiro valor inteiro dentre todos os valores com esse comprimento. O valor é calculado com base na seguinte regra:

$$\text{Base}[c] = \begin{cases} 0 & \text{se } c = 1, \\ 2 \times (\text{Base}[c - 1] + w_{c-1}) & \text{caso contrário,} \end{cases}$$



Regra para obter o valor do vetor Base

O vetor Offset indica o índice no vocabulário da primeira palavra de um determinado comprimento c . Quando não existe a palavra com o comprimento c , o último índice é repetido.

C	Base [c]	Offset[c]
1	0	1
2	2	2
3	6	2
4	12	3

Tabela gerada a partir dos vetores base e offset

A partir desses dois vetores e o vetor gerado pelo algoritmo de Moffat e Katajainen, eu pego **o tamanho do código** da palavra desejada, e descubro a **base e o offset do primeiro** valor de código com tamanho c. Dessa forma, eu aplico a **propriedade** que diz que códigos com tamanhos **iguais** são números inteiros consecutivos. Como já tenho a posição que eu desejo, eu somo a distância que a palavra desejada está da primeira no valor de base e converto esse valor para binário.

Para fazer a decodificação, ou seja, tendo o código, eu quero descobrir a palavra, basta eu **pegar o binário** e converter para decimal, somar com o offset e subtrair a base, dessa forma, eu vou ter o valor de i, que é o índice da palavra que referencia o código passado inicialmente.

COMPRESSÃO

A compressão é um processo demorado que possui três etapas:

- Cálculo do vocabulário e calcular as ocorrências de cada palavra. Para isso, o arquivo texto é percorrido e o vocabulário é gerado juntamente com a frequência
- Com base no resultado obtido, o vetor de vocabulário é ordenado em ordem decrescente e o algoritmo de Moffat e Katajainen é aplicado. Depois disso, os vetores base, offset e vocabulário são construídos e gravados no cabeçalho do arquivo comprimido.
- O arquivo texto é novamente percorrido, as palavras são extraídas e codificadas. Os códigos correspondentes são gravados no arquivo comprimido.

DESCOMPRESSÃO

O processo de descompressão é um processo mais rápido. Dado um arquivo comprimido, eu tenho como **objetivo gerar o arquivo texto correspondente**. Primeiramente realiza a **leitura** dos vetores base, offset e vocabulário gravados no início do arquivo comprimido. Depois, faz a **leitura dos códigos** do arquivo comprimido, descodificando-os e **gravando** as palavras correspondentes no arquivo texto. Como é uma etapa muito simples, ela é mais rápida, o que é ideal para a descompressão, pois não deixa o usuário esperando por muito tempo ao obter os resultados desejados.