Técnicas de Compactação e Compressão

Profa. Débora Christina Muchaluat Saade

debora@midiacom.uff.br

Técnicas de Compactação e Compressão

- ✓ **Compactação X Compressão**
- ✓ **Classificação das técnicas de compressão**
  - ***Codificação por Entropia, na Origem e Híbrida***
- ✓ **Técnicas de Compactação**
  - ***Codificação por carreira***
  - ***Codificação por Shannon-Fano***
  - ***Codificação de Huffman***
  - ***Codificação de Lempel-Ziv-Welch (LZW)***
  - ***Codificação aritmética***

Técnicas de Compactação

Profa. Débora Christina Muchaluat Saade

debora@midiacom.uff.br

Codificação de Huffman

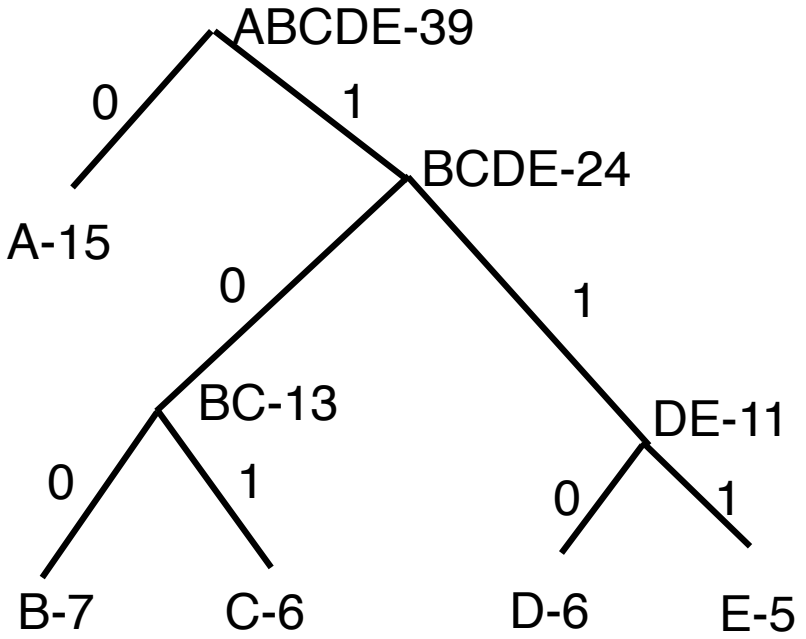
- ✓ **Codificação Estatística**
- ✓ **A sequência a ser compactada deve ser analisada previamente, identificando-se os caracteres e suas respectivas frequências/probabilidades**
- ✓ **Atribui menos bits a símbolos que aparecem mais frequentemente e mais bits para símbolos que aparecem menos**

Codificação de Huffman

- **Algoritmo:**
  - ***Insira os nós (símbolos/frequências) em uma lista***
  - ***Repita até que a lista contenha apenas um nó:***
    - Selecione os dois nós de mais baixa frequência e crie um nó pai para ambos
    - Atribua ao nó pai a soma das frequências e insira-o na lista
    - Atribua os códigos 0 e 1 aos dois ramos da árvore e retire os filhos da lista

Exemplo – Codificação de Huffman

- ✓ **Frequência dos símbolos:**
  - ***Símbolo – Frequência – Código – Número de bits***
  - ***Passo 1 - A15; B7; C6; D6; E5***
  - ***Passo 2 - A15; DE11; B7; C6***
  - ***Passo 3 - A15; BC13; DE11***
  - ***Passo 4 - BCDE24; A15***
  - ***Passo 5 - ABCDE39***

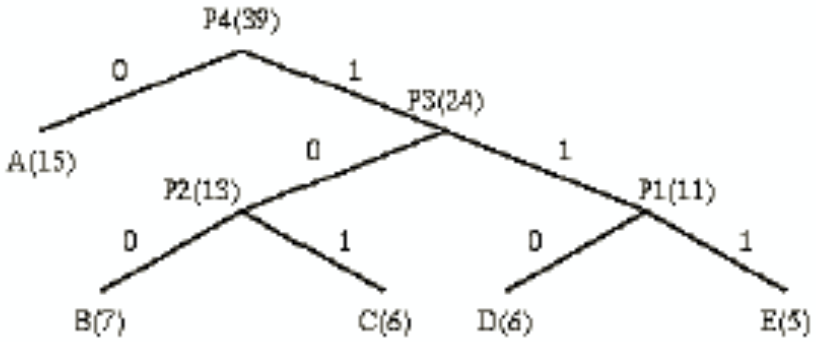


Exemplo - Huffman

Símbolo	Freqüência	Código	Subtotal (# de bits)
A	15	0	15
B	7	100	21
C	6	101	18
D	6	110	18
E	5	111	15

- ✓ **Sequência ABDEACD...**
  - *0 100 110 111 0 101 110 ...*
- ✓ **Taxa de Compactação**
  - *(39 símbolos x 3 bits) / (15+21+18+18+15) = 117:87*

Total (# de bits) = 87
Compactação = 117 : 87

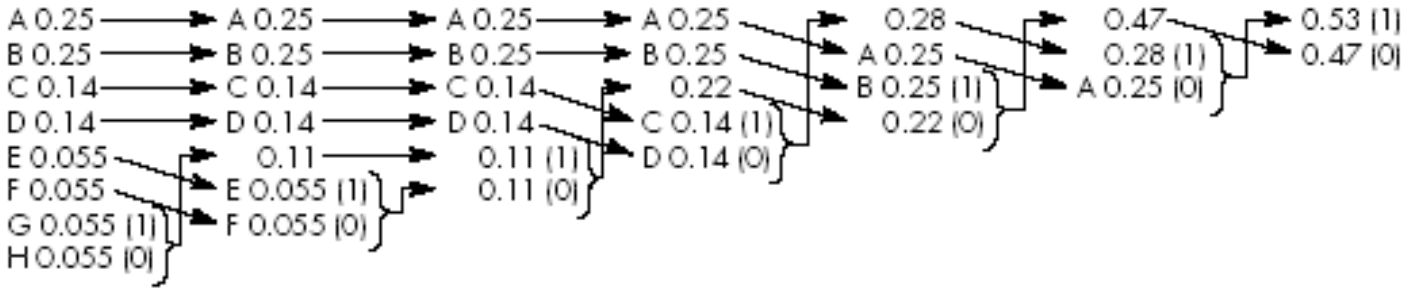


Exemplo - Huffman

- **Encontre os códigos de Huffman dos seguintes símbolos com as respectivas probabilidades de ocorrência:**
  - *A – 0,25*
  - *B – 0,25*
  - *C – 0,14*
  - *D – 0,14*
  - *E – 0,055*
  - *F – 0,055*
  - *G – 0,055*
  - *H – 0,055*

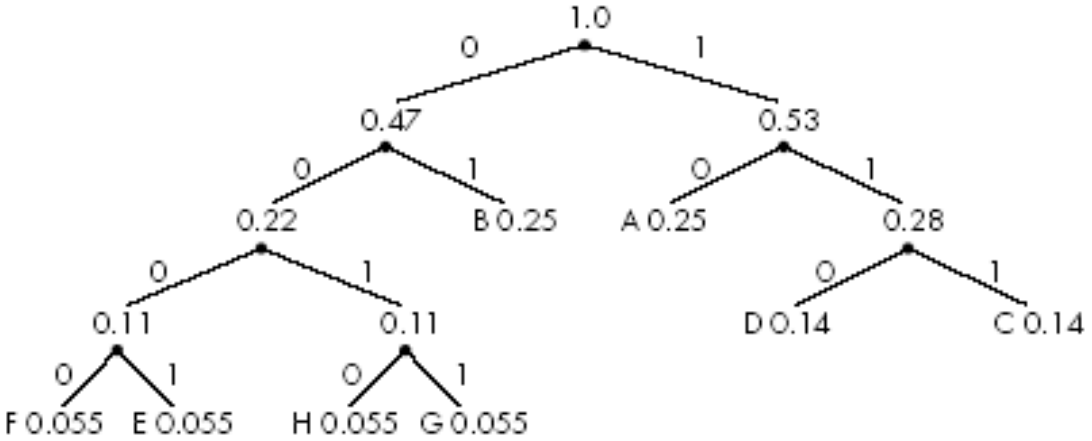
Exemplo - Huffman

(a)



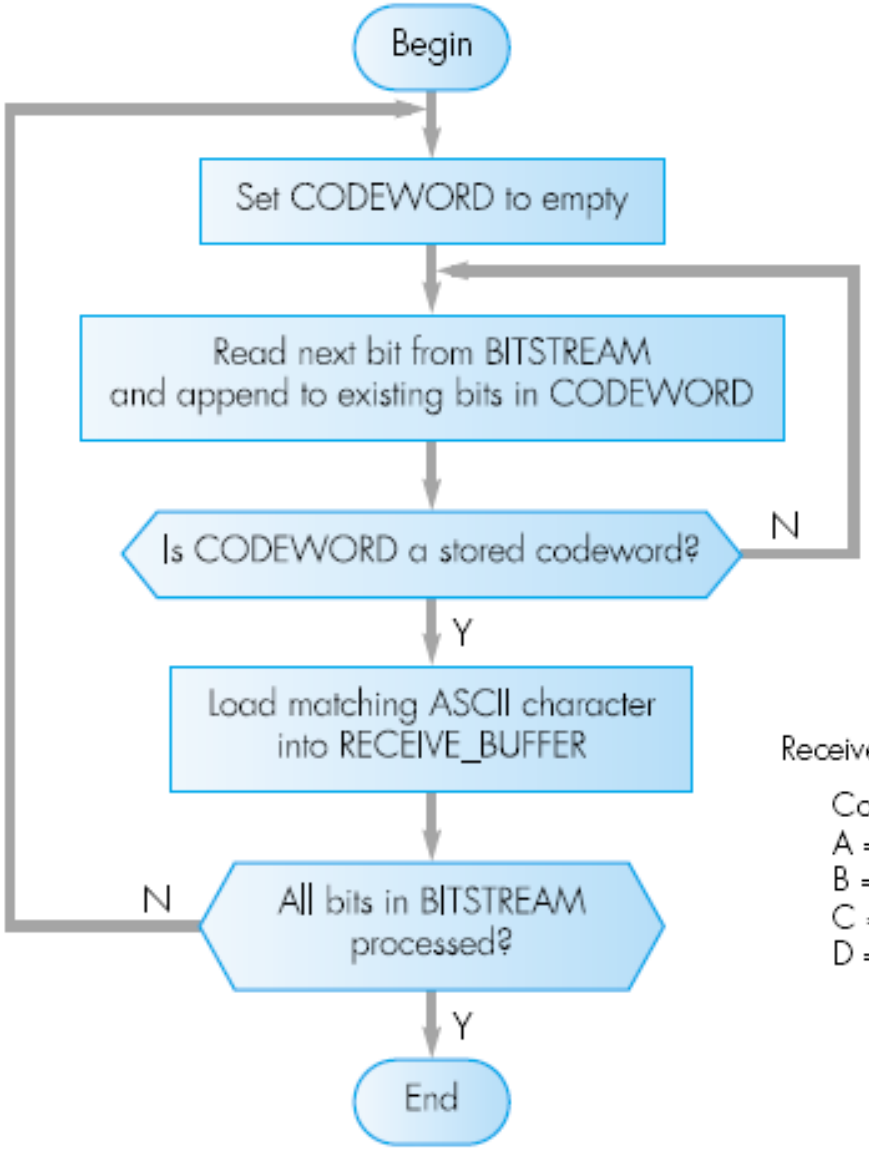
A = (0) (1) → 10
B = (1) (0) → 01
C = (1) (1) (1) → 111
D = (0) (1) (1) → 110
E = (1) (0) (0) (0) → 0001
F = (0) (0) (0) (0) → 0000
G = (1) (1) (0) (0) → 0011
H = (0) (1) (0) (0) → 0010

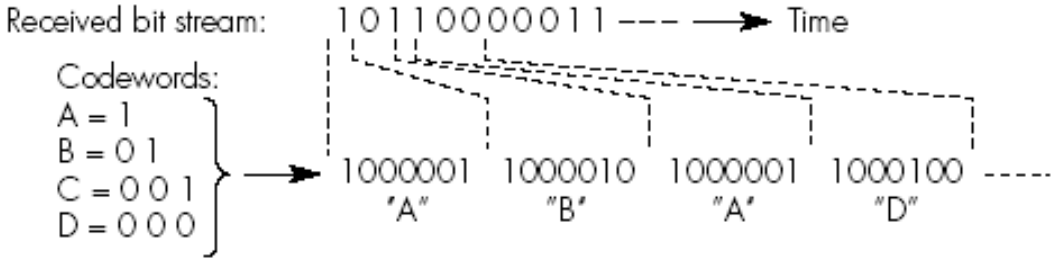
(b)



Weight order = 0.055 0.055 0.055 0.055 0.11 0.11 0.14 0.14 0.22 0.25 0.25 0.28 0.47 0.53 ✔

Huffman - Decodificação





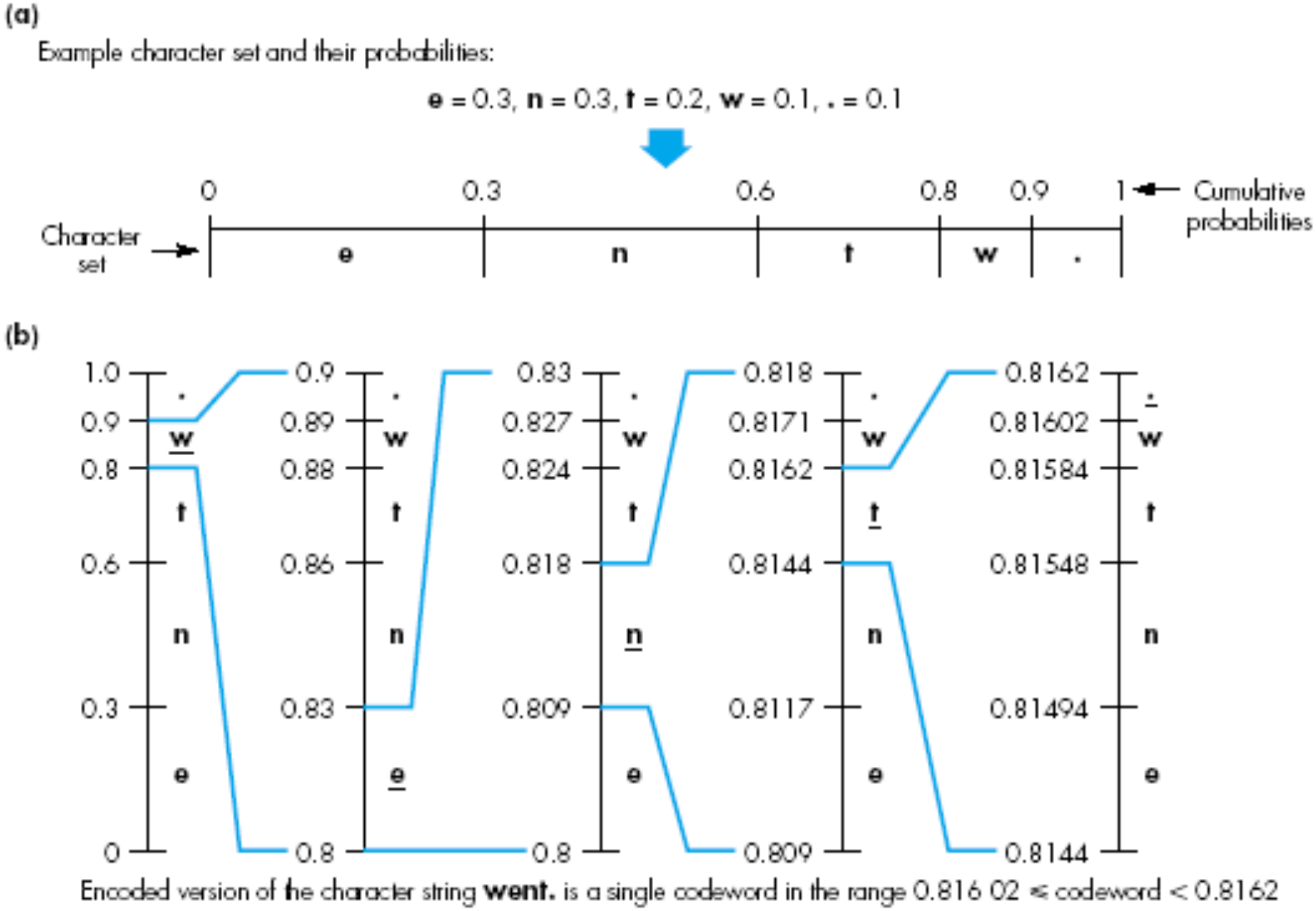
Codificação de Huffman

- ✓ **Nem todos os caracteres precisam ter uma representação codificada na tabela de Huffman**
  - ***apenas os caracteres com alta probabilidade de ocorrência***
  - ***demais são codificados diretamente e marcados com flag especial***
- ✓ **Técnica útil quando o número de caracteres diferentes é muito grande mas apenas alguns deles têm uma alta probabilidade de ocorrência**

Codificação Aritmética

- ✓ **Utiliza um único código por string de caracteres codificada**
- ✓ **A sequência a ser compactada deve ser analisada previamente, identificando-se os caracteres e suas respectivas frequências/ probabilidades**
  - *Ex.: e=0.3; n=0.3; t=0.2; w=0.1; .=0.1*
- ✓ **Precisa de um caracter marcando o fim de cada sequência (.)**
- ✓ **Algoritmo:**
  - ***Dividir o intervalo de [0, 1] de acordo com a probabilidade de ocorrência de cada caracter na sequência***
  - ***Repita até o caracter de fim de sequência***
    - Escolha o intervalo correspondente ao próximo caracter e divida-o novamente de acordo com as probabilidades iniciais
  - ***O código pode ser qualquer número dentro do último intervalo encontrado (ValorInicial <= código < ValorFinal)***

Exemplo – Codificação Aritmética



Encoded version of the character string **went.** is a single codeword in the range 0.816 02 ≤ codeword < 0.8162

Codificação Aritmética

- **O número de dígitos decimais no código aumenta linearmente com o número de caracteres na string**
  - ***O número máximo de caracteres em uma string é determinado pela precisão com a qual números de ponto flutuante são representados nos computadores de origem e destino***
  - ***Por essa razão, mensagens completas devem ser fragmentadas em várias strings menores e cada string deve ser codificada separadamente***

Codificação de Lempel-Ziv-Welch

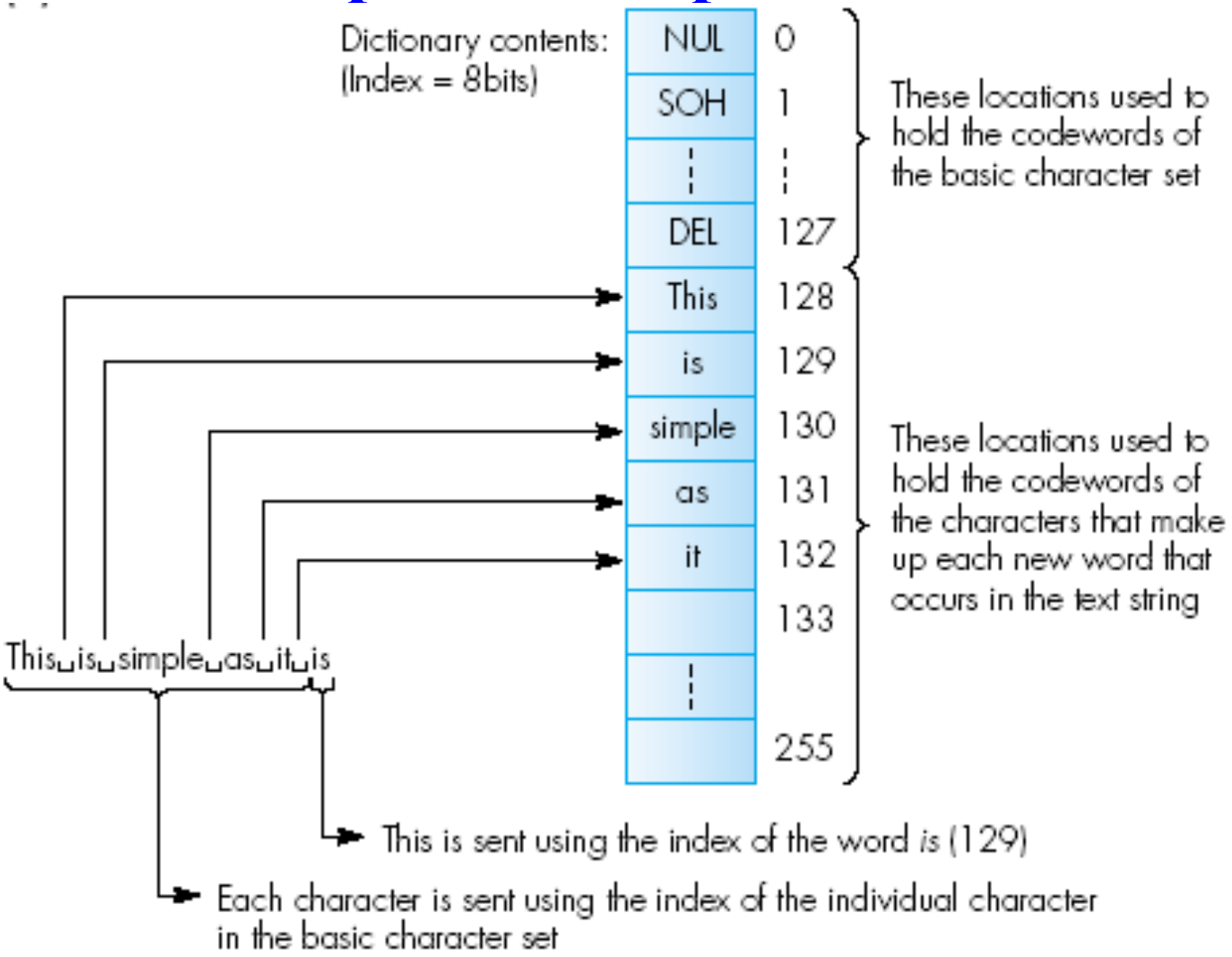
✓ Codificação de Lempel-Ziv-Welch (LZW)

- ***Ao invés de utilizar caracteres como a base da codificação, utiliza strings de caracteres***
- ***É baseada na construção de um dicionário de palavras (grupos de um ou mais caracteres) a partir do fluxo de entrada***
- ***Quando uma nova palavra é encontrada na sequência de entrada***
  - Ela é adicionada ao dicionário
- ***Se a palavra encontrada na sequência de entrada já foi registrada***
  - ela é substituída pelo código no dicionário
- ***Esta técnica é boa para compressão de arquivos textos, onde temos uma grande repetição de palavras de uso frequente***

Codificação de Lempel-Ziv-Welch (LZW)

- ✓ **Codificador e decodificador constroem o conteúdo do dicionário dinamicamente enquanto o texto é processado**
- ✓ **Inicialmente, o dicionário contém somente o conjunto de caracteres que foi usado na construção do texto (ex. ASCII 7 bits)**
- ✓ **As entradas restantes são utilizadas para codificar palavras que ocorrem no texto**
- ✓ **Exemplo:**
  - ***This is simple as it is ...***

Exemplo de Lempel-Ziv-Welch



Exemplo de Lempel-Ziv-Welch