Capítulo III

Códigos para Compressão sem Perdas

Vimos no Capítulo II que a saída Quantizador do Codificador de Fonte de um sistema para Transmissão Digital é codificada em seqüências de $N = \log_2 M$ bits, sendo M o número de níveis de quantização. Tal codificação é denominada PCM (PCM – *Pulse Code Modulation*), e a seqüência de bits resultante é denominada de seqüência PCM.

Neste capítulo é estudado o processo de codificação da seqüência de bits PCM, através do qual cada uma das M possíveis seqüências PCM de $N = \log_2 M$ bits têm o seu número de bits N reduzido para um valor menor como decorrência da ação do código para compressão de dados, sem que ocorra perda de informação. Já estudamos no Capítulo II códigos para compressão com base em critérios psico-acústicos, quais sejam, os algoritmos μ-*Law* e A-*Law*. Neste capítulo estudaremos dois códigos compressores propriamente ditos: O Código de Huffman (Codificação por Entropia) e o Algoritmo Lempel-Ziv (Codificação Universal de Fonte – Codificação por Dicionário).

3.1 Entropia – Uma Possível Medida de Informação

O conceito de informação é muito amplo para que possa ser capturado completamente por uma simples definição. No entanto, se estivermos tratando de uma variável aleatória discreta X, com espaço de amostras definido pelo conjunto $\Omega_X = \{m_k\} = \{m_0, m_1, \cdots, m_{M-1}\}$ de M eventos m_k com probabilidade de ocorrência p_k, $k = 0,1,\cdots, M-1$, é possível definir uma quantidade chamada de Entropia a qual apresenta muitas propriedades que concordam com noção intuitiva do que uma medida de informação deve expressar.

Intuitivamente, a observação da ocorrência de um evento do espaço amostral de uma variável aleatória nos dá informação. Eventos raros contém mais informação do que eventos comuns.

Por exemplo, nós inferimos e aprendemos muito pouco se alguém disser "O sol nasceu hoje pela manhã" ou disser "O sol nasceu à leste hoje pela manhã". Mas inferimos e aprendemos consideravelmente mais se alguém disser "São Paulo foi atingido por um terremoto hoje pela manhã" ou "São Paulo foi atingido por um furacão hoje pela manhã".

Em 1928 Hartley propôs uma medida logarítmica de medida de informação que reflete o raciocínio intuitivo do parágrafo anterior. Vamos supor que estamos registrando o valor das amostras na saída X do quantizador do Codificador de Fonte e que o quantizador apresente M níveis de quantização. Após o registro de um número suficiente de amostras é feito um estudo estatístico da probabilidade de ocorrência de cada uma das M possíveis amostras (ou mensagens de $N = \log_2 M$ bits sob o ponto de vista do código compressor), conforme discutido na Seção 1.2. A saída X do quantizador pode ser considerada uma variável aleatória discreta X, com espaço de amostras definido pelo conjunto

$\Omega_X = \{m_k\} = \{m_0, m_1, \cdots, m_{M-1}\}$ de *M* mensagens m_k com probabilidade de ocorrência p_k, $k = 0,1,\cdots, M-1$. Segundo Hartley, a Auto-Informação $h(m_k)$ implícita na ocorrência de uma mensagem m_k com probabilidade de ocorrência p_k, é definida por

$$h(m_k) = -\log_2(p_k) \ [bits] \tag{3.1}$$

Analisemos (3.1): Visto que $0 \le p_k \le 1$, $h(m_k)$ é sempre um número positivo. $h(m_k)$ é medida em [*bits*] devido à função logarítmica em base 2. Como $\log_2(u)$ é uma função monotonamente crescente com *u*, a Auto-Informação $h(m_k) = -\log_2(p_k)$ de uma mensagem rara é maior que a de uma mensagem comum. Ainda, a derivada $\frac{d}{dp_k}(-\log_2(p_k)) = \frac{-1}{p_k \ln(2)}$ é uma medida da sensibilidade de $h(m_k)$ à variação da probabilidade p_k de ocorrência da mensagem m_k. Observe, portanto, que a Auto-Informação varia pouco entre mensagens m_k comuns mas apresenta alta variação entre mensagens raras. Finalmente, observe que as propriedades de (3.1) concordam com noção intuitiva de informação discutida em parágrafo anteriores.

A média da Auto-Informação das *M* mensagens m_k do conjunto $\Omega_X = \{m_0, m_1, \cdots, m_{M-1}\}$ é denominada de Entropia da variável aleatória *X* (≡ Entropia do conjunto Ω_X de mensagens). Assim, a Entropia $H(X)$ da variável aleatória *X* cujo espaço de amostras é o conjunto Ω_X de *M* mensagens é dada por

$$H(X) = E\{h(m_k)\} = E\{-\log_2(p_k)\} = -\sum_{k=0}^{M-1} p_k \log_2(p_k) \ [bits] \tag{3.2}$$

onde o $E\{.\}$ é o operador estatístico que retorna o valor esperado do argumento [Carlson]. Note em (3.2) que, se as *M* mensagens apresentam probabilidade de ocorrência iguais (mensagens equiprováveis), então $p_k = 1/M$ para $k = 0,1,\cdots, M-1$ e $H(X) = -\frac{1}{M}\sum_{k=0}^{M-1} \log_2\left(\frac{1}{M}\right) = \log_2(M)$ [*bits*].

Além da interpretação da Entropia $H(X)$ como sendo a informação média implícita no conjunto de mensagens $\Omega_X = \{m_0, m_1, \cdots, m_{M-1}\}$, conjunto que é o espaço de amostras da variável aleatória *X*, são válidas também as seguintes interpretações:

1- A informação média obtida como resultado da observação de uma realização da variável aleatória *X* (realização = ocorrência de uma mensagem m_k na saída do quantizador). Nesta interpretação, $H(X)$ é melhor quantificada na unidade [*bits*/*realização*], ou no caso da saída do quantizador, em [*bits*/*mensagem*], ou mais genericamente, em [*bits*/*símbolo*].

2- A incerteza média sobre *X* antes de ocorrer uma observação.

3- A incerteza média sobre *X* removida após ocorrer uma observação.

Exemplo 3.1: Seja um sistema para transmissão digital que utilize no Codificador de Fonte um conjunto $\Omega_X = \{m_0, m_1\}$ com $M = 2$ possíveis mensagens (ou $M = 2$ níveis de quantização sob o ponto de vista do quantizador). Seja q a probabilidade de que a saída X do quantizador assuma o valor m_0, isto é, $q = P(X = m_0)$. Determine o gráfico da entropia de X em função de q.

Solução: Se $q = P(X = m_0)$, então $P(X = m_1) = 1 - q$. De (3.2) temos

$$H(X) = -q \log_2 q - (1-q)\log_2(1-q) \ [bits/mensagem] \tag{3.3}$$

A Figura 3.1 mostra o gráfico $H(X) \times q$.

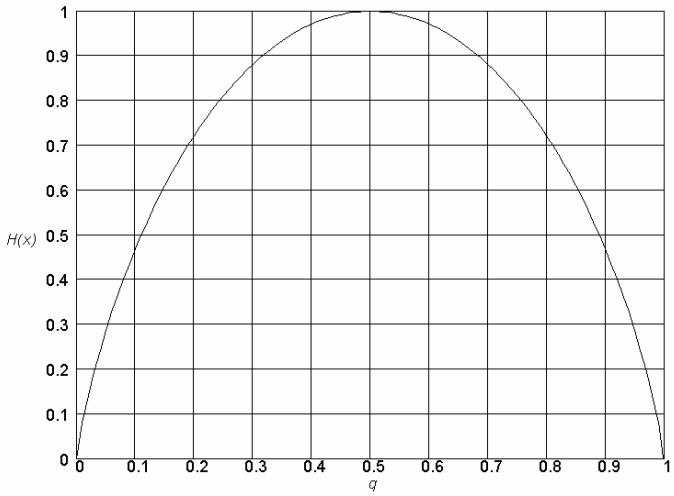

Figura 3.1: Entropia de X em função de q.

[F1] Comentário: F2_1_EntropBin.pcx

Note na Figura 3.1 que $H(X)$ é máxima quando as mensagens m_0 e m_1 têm a mesma probabilidade de ocorrência, i.e., $q = (1-q) = 0.5$. Na realidade, este comportamento acontece não só para um espaço de amostras Ω_X com apenas $M = 2$ mensagens de probabilidades iguais, mas ocorre também para qualquer quantidade M de mensagens de mesma probabilidade. Demonstra-se [Ash] que

O valor máximo da entropia da variável aleatória X é $H(X) = \log_2(M)$, valor que ocorre quando as probabilidades de ocorrência dos M elementos do espaço de amostras Ω_X são todas iguais a $1/M$ (i.e., os M elementos de Ω_X são equiprováveis).

3.2 Codificação por Entropia

Vimos o conceito de Entropia como uma medida do conteúdo de informação associado a uma variável aleatória discreta X, com espaço de amostras definido pelo conjunto $\boldsymbol{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ de M eventos x_i com probabilidade de ocorrência p_i, $i = 0,1,\cdots, M-1$.

Quando *X* é a saída de uma fonte de informação discreta, (por exemplo, a saída do Quantizador do Codificador de Fonte) a entropia $H(X)$ da fonte representa a quantidade média de informação emitida pela fonte.

Podemos considerar um código para compressão por entropia como um operador $\boldsymbol{\theta}\{\cdot\}$, tal que $\mathbf{S} = \boldsymbol{\theta}\{\boldsymbol{\Omega}\}$, onde $\boldsymbol{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ é o conjunto de *M* possíveis **mensagens** a serem codificadas e $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ é o conjunto de *M* possíveis **palavras-código** ou **símbolos** resultantes da codificação. O operador $\boldsymbol{\theta}\{\cdot\}$ efetua um mapeamento unívoco entre cada mensagem e respectiva palavra-código tal que mensagens com maior probabilidade de ocorrência são mapeadas em palavras-código de menor tamanho, e mensagens com menor probabilidade de ocorrência são mapeadas em palavras-código de maior tamanho (no caso de um código binário, "tamanho" refere-se ao número de bits). O **conjunto de caracteres do código** ou **alfabeto do código** é o conjunto $\mathbf{A} = \{a_0, a_1, \cdots, a_{D-1}\}$ composto por *D* elementos, de cuja composição são formadas cada palavra-código. As palavras-código formadas do alfabeto $\mathbf{A}$, as quais constituem o **conjunto imagem** do mapeamento $\boldsymbol{\theta}\{\cdot\}$, são assumidas serem distintas entre si, caso contrário $\boldsymbol{\theta}\{\cdot\}$ não seria unívoco.

Exemplo 3.2: Seja o alfabeto $\mathbf{A} = \{a_0, a_1, a_2\}$ e o conjunto de mensagens $\boldsymbol{\Omega} = \{x_0, x_1, x_2, x_3\}$. Um possível código $\boldsymbol{\theta}\{\cdot\}$ seria

Mensagem	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	$a_0 a_1$
x_1	$a_0 a_1 a_2$
x_2	a_0
x_3	a_1

No contexto de Codificação de Fonte em Transmissão Digital, as mensagens são seqüências de bits resultantes de codificação PCM da saída do Quantizador.

Exemplo 3.3: Seja o alfabeto $\mathbf{A} = \{a_0, a_1, a_2\}$ e o conjunto de mensagens $\boldsymbol{\Omega} = \{x_0, x_1, x_2, x_3\} = \{00,01,10,11\}$ resultante da codificação PCM da saída de um Quantizador com 4 níveis de quantização. Um possível código $\boldsymbol{\theta}\{\cdot\}$ seria

Mensagem	Seqüência PCM	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	00	$a_0 a_1$
x_1	01	$a_0 a_1 a_2$
x_2	10	a_0
x_3	11	a_1

Ainda no contexto de Transmissão Digital, as palavras-código usualmente originam-se de um alfabeto binário $\mathbf{A} = \{0,1\}$.

Exemplo 3.4: Seja o alfabeto $\mathbf{A} = \{0,1\}$ e o conjunto de mensagens $\mathbf{\Omega} = \{x_0, x_1, x_2, x_3\} = \{00,01,10,11\}$. Um possível código $\boldsymbol{\theta}\{\cdot\}$ seria

Mensagem	Seqüência PCM	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	00	0
x_1	01	010
x_2	10	01
x_3	11	10

O **tamanho** ℓ_i de uma palavra-código ou símbolo s_i é definido pelo número de caracteres do alfabeto **A** utilizado na sua construção.

Exemplo 3.5: Seja o código binário ($\mathrm{A} = \{0,1\}$) do Exemplo 3.4. O tamanho ℓ_i de cada palavra-código ou símbolo s_i é

Mensagem	Seqüência PCM	Símbolo s_i associado a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$	ℓ_i
x_0	00	0	1
x_1	01	010	3
x_2	10	01	2
x_3	11	10	2

O objetivo da Codificação por Entropia é encontrar um código $\boldsymbol{\theta}\{\cdot\}$ que minimize o tamanho médio $\overline{L}$ dos símbolos emitidos pela fonte a partir do conjunto de *M* possíveis símbolos $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$, sendo $\overline{L}$ dado por

$$\overline{L} = \sum_{i=0}^{M-1} p_i \ell_i \tag{3.4}$$

onde p_i é a probabilidade de ocorrência da mensagem x_i, e ℓ_i é o tamanho do símbolo s_i associado à mensagem x_i através do código $\boldsymbol{\theta}\{\cdot\}$.

A Codificação por Entropia assume que a fonte é **sem memória**. Uma fonte é considerada sem memória quando as mensagens emitidas pela fonte são estatisticamente independentes, i.e., a ocorrência de uma determinada mensagem x_i não afeta a probabilidade de ocorrência da mensagem x_j, com $i, j = 0,1,\cdots,M-1$. Esta condição é necessária pois caso contrário a função $\overline{L} = f(p_i, \ell_i)$ a ser minimizada, dada por (3.4), dependeria do desenrolar temporal da seqüência de mensagens emitidas pela fonte, o que

resultaria em um código $\boldsymbol{\theta}\{.\}$ variável no tempo. Embora poucas fontes físicas sigam exatamente o modelo de uma fonte sem memória, códigos $\boldsymbol{\theta}\{.\}$ constantes no tempo (resultantes da suposição de independência estatística) são amplamente utilizados como códigos compressores mesmo quando a dependência estatística da fonte resulta na impossibilidade de minimização de $\bar{L}$ durante a totalidade do tempo de codificação.

Exemplo 3.6: Seja um sistema para transmissão digital que utilize no Codificador de Fonte um conjunto $\boldsymbol{\Omega} = \{x_0, x_1, x_2, x_3\} = \{00,01,10,11\}$ com $M = 4$ possíveis mensagens (ou $M = 4$ níveis de quantização sob o ponto de vista do quantizador). As amostras na saída *X* do quantizador são tais que a ocorrência de uma não altera a probabilidade de ocorrência da outra (i.e., as mensagens são estatisticamente independentes). As probabilidades são $P(X = x_0) = 1/2$, $P(X = x_1) = 1/4$ e $P(X = x_2) = P(X = x_3) = 1/8$. O código compressor $\boldsymbol{\theta}\{.\}$ é conforme abaixo

Mensagem	Seqüência PCM	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	00	0
x_1	01	10
x_2	10	110
x_3	11	111

Determine a entropia da saída do quantizador $H(X)$ e o comprimento médio $\bar{L}(\boldsymbol{\theta})$ do código $\boldsymbol{\theta}\{.\}$.

Solução:

Mensagem	p_i	Símbolo s_i associado a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$	ℓ_i
x_0	$1/2$	0	1
x_1	$1/4$	10	2
x_2	$1/8$	110	3
x_3	$1/8$	111	3

$$H(X) = -\frac{1}{2}\log_2\left(\frac{1}{2}\right) - \frac{1}{4}\log_2\left(\frac{1}{4}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) - \frac{1}{8}\log_2\left(\frac{1}{8}\right) = 1.75 \ \left[bits/mensagem\right] \quad (3.5)$$

$$\bar{L}(\boldsymbol{\theta}) = \frac{1}{2}\times 1 + \frac{1}{4}\times 2 + \frac{1}{8}\times 3 + \frac{1}{8}\times 3 = 1.75 \ \left[bits/símbolo\right] \quad (3.6)$$

Exemplo 3.7: Seja o código código compressor $\boldsymbol{\theta}\{\cdot\}$ é conforme abaixo

Mensagem	p_i	Símbolo s_i associado a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	1/3	0
x_1	1/3	10
x_2	1/3	11

Determine a entropia $H(X)$ da fonte e o comprimento médio $\bar{L}(\boldsymbol{\theta})$ do código $\boldsymbol{\theta}\{\cdot\}$.

Solução:

$$H(X) = -\frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) - \frac{1}{3}\log_2\left(\frac{1}{3}\right) = 1.58 \ \left[bits/mensagen\right] \tag{3.7}$$

$$\bar{L}(\boldsymbol{\theta}) = \frac{1}{3}\times 1 + \frac{1}{3}\times 2 + \frac{1}{3}\times 2 = 1.67 \ \left[bits/símbolo\right] \tag{3.8}$$

3.2.1 Códigos Univocamente Decodificáveis

Um código que pretenda ser útil deve pertencer à classe de códigos Univocamente Decodificáveis, caso contrário é impossível efetuar a decodificação sem que ocorra ambigüidade.

Um código é Univocamente Decodificável (UD) quando qualquer seqüência de caracteres do alfabeto **A** passível de ser formada a partir da justaposição de um número qualquer de símbolos quaisquer pertencentes a $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ puder ser associada, ao ser decodificada, a uma única mensagem em $\boldsymbol{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$.

Conceito de justaposição: A justaposição de N símbolos (ou palavras-código) $s_i, s_{i+1}, \cdots, s_{i+N-1}$ é a seqüência α formada pela transmissão do símbolo s_i seguido da transmissão do símbolo s_{i+1}, e assim sucessivamente até a transmissão do símbolo s_{i+N-1}, cuja representação é $\alpha = s_i s_{i+1} \cdots s_{i+N-1}$.

Exemplo 3.8: Verifique se o código $\boldsymbol{\theta}\{\cdot\}$ abaixo é UD.

Mensagem	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	0
x_1	010
x_2	01
x_3	10

Solução: A seqüência 010 poderia corresponder a qualquer uma das seguintes seqüências de mensagens x_1, x_2x_0 ou x_0x_3. Portanto $\boldsymbol{\theta}\{\cdot\}$ não é UD.

3.2.2 Códigos Instantâneos (=Códigos Prefixos)

No Exemplo 3.8 a ambigüidade do código $\boldsymbol{\theta}\{\cdot\}$ talvez pudesse ser resolvida se aguardássemos a recepção de bits adicionais, mas tal tempo de espera é indesejável dada a sempre existente busca por velocidade de decodificação.

Uma maneira de assegurar que um código seja UD **e** que nenhum tempo de espera seja necessário para a correta decodificação é utilizar códigos denominados Prefixos ou Instantâneos (a denominação "Instantâneo" decorre da não necessidade de aguardar a recepção de bits adicionais para que se resolva ambigüidades). **Nota**: Todos os códigos instantâneos são UD, mas nem todos os códigos UD são instantâneos. Ou seja, o conjunto dos códigos instantâneos é um sub-conjunto do conjunto dos códigos UD.

Um código instantâneo ou prefixo pode ser decodificado sem referência a palavras-código futuras porque o final de uma palavra-código é imediatamente reconhecida no decodificador.

Um código é chamado Instantâneo se nenhuma palavra-código é prefixo de nenhuma outra palavra palavra-código pertencente ao código.

Conceito de prefixo: Sejam as seqüências α_a, α_b e α_c, formadas pela justaposição de respectivamente N_a, N_b e N_c palavras-código s_i pertencentes ao código $\boldsymbol{\theta}\{\cdot\}$, sendo $N_a = N_b + N_c$ um número qualquer de palavras-código. Dizemos que α_b é prefixo de α_a se α_a puder ser representada por $\alpha_b\alpha_c$, para alguma seqüência α_c denominada sufixo.

Exemplo 3.9: Verifique se o código $\boldsymbol{\theta}\{\cdot\}$ abaixo é Instantâneo.

Mensagem	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	10
x_1	00
x_2	11
x_3	110

Solução: Como 11 é prefixo de 110 , $\boldsymbol{\theta}\{\cdot\}$ não é Instantâneo. Não podemos afirmar que não seja UD pelo fato de não ser Instantâneo.

3.2.3 Procedimento geral para testar se um código é UD

Seja um código $\boldsymbol{\theta}\{\cdot\}$ com alfabeto $\mathbf{A} = \{a_0, a_1, \cdots, a_{D-1}\}$ e conjunto imagem $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$. Para testar se $\boldsymbol{\theta}\{\cdot\}$ é UD, constrói-se a seqüência de conjuntos $\mathbf{S}_0, \mathbf{S}_1, \cdots$ da seguinte maneira:

$\mathbf{S}_0$ é o próprio conjunto imagem $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$.

Para definir $\mathbf{S}_1$, forma-se a partir de $\mathbf{S}_0$ o conjunto $\mathbf{P}$ de todos os pares $s_i s_j$ de palavras-código, $s_i \neq s_j$, possíveis de serem formados por justaposição de duas palavras-código distintas pertencentes ao conjunto $\mathbf{S}_0$:

	s_0	s_1	$\cdots$	s_{M-1}
s_0	-	$s_0\ s_1$	$\cdots$	$s_0\ s_{M-1}$
s_1	$s_1\ s_0$	-	$\cdots$	$s_1\ s_{M-1}$
$\vdots$	$\vdots$	$\vdots$	-	$\vdots$
s_{M-1}	$s_{M-1}\ s_0$	$s_{M-1}\ s_1$	$\cdots$	-

Tabela 3.1: Formação do conjunto $\mathbf{P} = \{s_0 s_1, s_0 s_2, \cdots, s_{M-1} s_{M-2}\}$ de $M^2 - M$ elementos.

Se a palavra-código $s_i \in \mathbf{S}_0$ é prefixo da palavra-código $s_j \in \mathbf{S}_0$, i.e. $s_j = s_i \sigma$, então o sufixo σ é um elemento do conjunto $\mathbf{S}_1$, i.e. $\sigma \in \mathbf{S}_1$. Executa-se a verificação $s_j \overset{?}{=} s_i \sigma$ para todos os elementos de $\mathbf{P}$ até que todos os sufixos sejam atribuídos ao conjunto $\mathbf{S}_1 = \{\alpha_0, \alpha_1, \cdots\}$, onde cada seqüência α_k de caracteres de A é um sufixo originado pelo resultado positivo do teste $s_j \overset{?}{=} s_i \sigma$.

Para definir $\mathbf{S}_n$, $n > 1$, compara-se $\mathbf{S}_0$ e $\mathbf{S}_{n-1}$ de modo bidirecional: **I)** Se uma palavra-código $s_i \in \mathbf{S}_0$ é prefixo de uma seqüência $\alpha_j \in \mathbf{S}_{n-1}$ tal que $\alpha_j = s_i \sigma$ então o sufixo $\sigma \in \mathbf{S}_n$. **II)** Se uma seqüência $\alpha'_j \in \mathbf{S}_{n-1}$ é prefixo de uma palavra-código $s'_i \in \mathbf{S}_0$ tal que $s'_i = \alpha'_j \sigma'$ então o sufixo $\sigma' \in \mathbf{S}_n$.

Define-se tantos conjuntos $\mathbf{S}_n$ até um valor de *n* tal que $\mathbf{S}_n = \{\Phi\}$ ou até um valor de *n* tal que $\mathbf{S}_n = \mathbf{S}_{n-1}$.

O código $\boldsymbol{\theta}\{\cdot\}$ é UD se e somente se **nenhum** dos conjuntos da seqüência de conjuntos $\mathbf{S}_1, \mathbf{S}_2, \cdots$ contenha uma palavra-código que pertença ao conjunto $\mathbf{S}_0$.

Exemplo 3.10: Verifique se o código $\boldsymbol{\theta}\{\cdot\}$ abaixo com alfabeto $\mathbf{A} = \{a, b, c, d, e\}$ é UD.

Mensagem	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	*a*
x_1	*c*
x_2	*ad*
x_3	*abb*
x_4	*bad*

x_5	*deb*
x_6	*bbcde*

Solução:

$\mathbf{S}_0$	$\mathbf{S}_1$	$\mathbf{S}_2$	$\mathbf{S}_3$	$\mathbf{S}_4$	$\mathbf{S}_5$	$\mathbf{S}_6$	$\mathbf{S}_7$	$\mathbf{S}_8$
a	*d*	*eb*	*de*	*b*	*ad*	*d*	*eb*	$\{\Phi\}$
c	*bb*	*cde*			*bcde*			
ad								
abb								
bad								
deb								
bbcde								

Visto que $ad \in \mathbf{S}_5$ e $ad \in \mathbf{S}_0$, logo $\boldsymbol{\theta}\{\cdot\}$ não é UD. Note que poderíamos ter encerrado o procedimento ao obter $\mathbf{S}_5$, quando, então, já temos elementos suficientes para decidir que $\boldsymbol{\theta}\{\cdot\}$ não é UD.

Exemplo 3.11: Verifique se o códigos $\boldsymbol{\theta}_{\text{I}}\{\cdot\}$, $\boldsymbol{\theta}_{\text{II}}\{\cdot\}$ e $\boldsymbol{\theta}_{\text{III}}\{\cdot\}$ abaixo são UD e/ou Instantâneos.

Mensagem	$s_i = \boldsymbol{\theta}_{\text{I}}\{x_i\}$	$s_i = \boldsymbol{\theta}_{\text{II}}\{x_i\}$	$s_i = \boldsymbol{\theta}_{\text{III}}\{x_i\}$
x_0	1	0	0
x_1	00	10	01
x_2	01	110	011
x_3	10	111	111

Solução:

Verificando $\boldsymbol{\theta}_{\text{I}}\{\cdot\}$:

$\boldsymbol{\theta}_{\text{I}}\{\cdot\}$ não é Instantâneo (1 é prefixo de 10), mas pode ser UD. Aplicando o procedimento da Seção 3.2.3:

$\mathbf{S}_0$	$\mathbf{S}_1$	$\mathbf{S}_2$
1	0	0
00		1
01		
10		

Visto que $1 \in \mathbf{S}_2$ e $1 \in \mathbf{S}_0$, $\boldsymbol{\theta}_{\text{I}}\{\cdot\}$ não é UD.

Verificando $\boldsymbol{\theta}_{\text{II}}\{\cdot\}$:

$\boldsymbol{\theta}_{\text{II}}\{\cdot\}$ é Instantâneo (nenhuma palavra-código é prefixo de nenhuma outra) então $\boldsymbol{\theta}_{\text{II}}\{\cdot\}$ é UD.

Apenas a título de ilustração vamos verificar se $\boldsymbol{\theta}_{\text{II}}\{\cdot\}$ é UD pelo procedimento da Seção 3.1.3:

$\mathbf{S}_0$	$\mathbf{S}_1$
0	$\{\Phi\}$
10	
110	
111	

Como nenhum dos conjuntos da seqüência de conjuntos $\mathbf{S}_1, \mathbf{S}_2, \cdots$ contém uma palavra-código que pertença ao conjunto $\mathbf{S}_0$, $\boldsymbol{\theta}_{\text{II}}\{\cdot\}$ é UD .

Verificando $\boldsymbol{\theta}_{\text{III}}\{\cdot\}$:

$\boldsymbol{\theta}_{\text{III}}\{\cdot\}$ não é Instantâneo (0 é prefixo de 01, por exemplo), mas pode ser UD. Aplicando o procedimento da Seção 3.2.3:

$\mathbf{S}_0$	$\mathbf{S}_1$	$\mathbf{S}_2$
0	1	11
01	11	1
011		
111		

Como nenhum dos conjuntos da seqüência de conjuntos $\mathbf{S}_1, \mathbf{S}_2, \cdots$ contém uma palavra-código que pertença ao conjunto $\mathbf{S}_0$, $\boldsymbol{\theta}_{\text{III}}\{\cdot\}$ é UD .

3.2.3 Teorema da Codificação de Fonte (*Noiseless Coding Theorem*)

Demonstra-se que [Ash][Cover]:

"Seja uma variável aleatória discreta *X*, com espaço de amostras definido pelo conjunto $\boldsymbol{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ de *M* eventos estatisticamente independentes x_i com probabilidade de ocorrência p_i, $i = 0,1,\cdots,M-1$. Então é possível construir um código Instantâneo $\boldsymbol{\theta}\{\cdot\}$ com um conjunto de palavras-código $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ formadas a partir do alfabeto $\mathbf{A} = \{0,1\}$ tal que o conjunto $\mathbf{L} = \{\ell_i\} = \{\ell_0, \ell_1, \cdots, \ell_{M-1}\}$ dos tamanhos das palavras-código respectivas em $\mathbf{S}$ satisfaz a desigualdade

$$H(X) \le \bar{L} < H(X) + 1 \tag{3.9}$$

onde $H(X)$ é a Entropia *X* da fonte e $\bar{L}$ é o tamanho médio das palavras-códigos dado por $\bar{L} = \sum_{i=0}^{M-1} p_i \ell_i$".

O Teorema da Codificação de Fonte (TCF) garante a viabilidade teórica de implementação de códigos instantâneos binários cujo tamanho médio dos símbolos pode

ser reduzido a um valor tão pequeno quanto o valor da Entropia $H(X)$ da fonte, ou, se impossível, pelo menos a um valor menor que $H(X)+1$.

3.2.4 Eficiência de um Código por Entropia

Uma decorrência natural do TCF é a definição da Eficiência de Codificação η dada por

$$\eta = \frac{H(X)}{\overline{L}} \tag{3.10}$$

Um código é **Absolutamente Ótimo** (*matched to the source* – casado com a fonte) quando $n = 1.0$, isto é, quando $H(X) = \overline{L}$.

Um código é **Quase Absolutamente Ótimo**, quando $H(X) \le \overline{L} < H(X)+1$.

Por exemplo, o código do Exemplo 3.6 é um código Absolutamente Ótimo.

3.2.5 Extensão da Fonte

No TCF, dado por (3.9), existe um descasamento residual fonte-código que pode chegar a quase 1 bit (termo à direita em (3.9)). Este descasamento residual ocorre devido ao fato que $\log_2 p_i$, usado no cálculo de $H(X)$, nem sempre é um número inteiro. Podemos reduzir este descasamento por palavra-código diluindo-o ao longo de várias palavras-código através da operação de Extensão da Fonte:

Seja um sistema de codificação no qual, ao invés de associarmos uma palavra-código $s_i \in \mathbf{S}$ a cada mensagem $x_i \in \mathbf{\Omega}$, tomamos uma seqüência de J observações independentes de X (uma "super mensagem") e atribuímos uma "super palavra-código" composta por J palavras-código s_i à seqüência de J mensagens x_i. Nesta situação o TCF pode ser re-escrito como [Cover]:

$$H(X) \le \frac{\overline{L}_{\mathrm{J}}}{\mathrm{J}} < H(X) + \frac{1}{\mathrm{J}} \tag{3.11}$$

onde $\overline{L}_{\mathrm{J}}/\mathrm{J} = \overline{L}$. Portanto, quanto maior o tamanho J das "super palavras-código" gerada por extensão de fonte menor o descasamento residual $1/\mathrm{J}$.

3.2.6 Códigos Ótimos – Códigos de Huffman

Embora o TCF nos garanta que é possível obter códigos instantâneos com $\overline{L}$ tão pequeno quanto a própria Entropia $H(X)$ da fonte, nenhuma informação é dada sobre **como** construir tais códigos.

A construção de tais códigos baseia-se na minimização de $\overline{L} = \sum_{i=0}^{M-1} p_i \ell_i$. Um código instantâneo que mimimize $\overline{L}$ é denominado de **Código Ótimo**.

Existe um teorema que prova que se um código ótimo $\boldsymbol{\theta}^*\{\cdot\}$ resulta em $\bar{L}^*$, é impossível existir um outro código instantâneo $\boldsymbol{\theta}\{\cdot\}$ com tamanho médio $\bar{L}$ tal que $\bar{L} < \bar{L}^*$ [Ash].

Um Código Ótimo binário cujas palavras-código $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ são formadas a partir do alfabeto $\mathbf{A} = \{0,1\}$ satisfaz as seguintes propriedades [Cover]:

1- Palavras-código com maior probabilidade possuem menor tamanho.

2- As duas palavras-código menos prováveis possuem o mesmo tamanho.

3- As duas palavras-código menos prováveis diferem somente no último bit.

Exemplo 3.12: Verifique se o código $\boldsymbol{\theta}\{\cdot\}$ abaixo é Ótimo.

Mensagem	p_i	Palavra-código s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	0.6	0
x_1	0.2	100
x_2	0.1	101
x_3	0.04	1101
x_4	0.06	1110

Solução:

As propriedades 1 e 2 são satisfeitas. No entanto a propriedade 3 não é satisfeita: x_3 e x_4 não diferem somente no último bit. Portanto, $\boldsymbol{\theta}\{\cdot\}$ não é ótimo.

O método de construção de códigos ótimos é devido a Huffman [Huffman]. Analisaremos aqui apenas o caso binário (Para o caso *D*-ário ver ***Codificação de Sinais*** por F.C.C De Castro e M.C.F. De Castro , **Capítulo III – Códigos para Compressão sem Perdas**, disponível para *download* em http://www.ee.pucrs.br/~decastro/download.html**).** Embora existam variantes do método que também geram Códigos Ótimos, adotaremos o seguinte procedimento para implementação de Códigos de Huffman:

Seja uma fonte de informação representada pela variável aleatória discreta *X*, com espaço de amostras definido pelo conjunto $\boldsymbol{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ de *M* eventos estatisticamente independentes x_i com probabilidade de ocorrência p_i, $i = 0,1,\cdots,M-1$. Seja $\boldsymbol{\theta}\{\cdot\}$ o Código de Huffman com palavras-código $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ formadas a partir do alfabeto $\mathbf{A} = \{0,1\}$ tal que o conjunto $\mathbf{L} = \{\ell_i\} = \{\ell_0, \ell_1, \cdots, \ell_{M-1}\}$ dos tamanhos das palavras-código respectivas em $\mathbf{S}$ minimiza $\bar{L} = \sum_{i=0}^{M-1} p_i \ell_i$. Para a construção de $\boldsymbol{\theta}\{\cdot\}$ efetua-se:

Seja, inicialmente, ***k***=***j***=0.

1- Organizar as probabilidades p_i de alto a baixo em uma coluna em ordem decrescente de valor, denominada Coluna ***k***.

2- Somar as duas menores probabilidades na Coluna ***k*** e transferi-las para a próxima coluna (à direita), denominada Coluna ***k*+1,** obedecendo a ordem decrescente. As demais probabilidades da Coluna ***k*** são transferidas inalteradas para a Coluna ***k*+1**.

3- Incrementar ***k*** de 1 e repetir 1 a 3 até restarem somente duas probabilidades na Coluna ***k*+1,** então denominada Coluna ***j***.

4- Na Coluna ***j*** , atribuir a palavra-código representada pelo caractere 0 à maior probabilidade e atribuir a palavra-código representada pelo caractere 1 à menor probabilidade

5- Localizar na Coluna ***j*+1**, imediatamente à esquerda da Coluna ***j***, quais as duas probabilidades geradoras que, ao serem somadas, resultaram na probabilidade gerada na Coluna ***j*.** Atribuir às duas probabilidades geradoras na coluna Coluna ***j*+1** a palavra-código já atribuída à probabilidade gerada na Coluna ***j*** . Às probabilidades não-geradoras na Coluna ***j*+1** são atribuídas as palavras-código já atribuídas às respectivas probabilidades não-geradas por soma na Coluna ***j*** .

6- Na Coluna ***j*+1**, às palavras-códigos já atribuídas em 5 às duas probabilidades geradoras justapor a palavra-código representada pelo caractere 0 àquela geradora de maior probabilidade, e justapor a palavra-código representada pelo caractere 1 àquela geradora de menor probabilidade.

7- Incrementar ***j*** de 1 e repetir 5 a 7 até que todas as colunas tenham palavras-código associadas às probabilidades nelas contidas.

8- Após a execução de 7, o Código de Huffman estará definido na coluna mais à esquerda.

Exemplo 3.13: Seja uma fonte de informação representada pela variável aleatória discreta *X*, com espaço de amostras definido pelo conjunto $\mathbf{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ de $M = 6$ eventos estatisticamente independentes x_i com probabilidade de ocorrência p_i, $i = 0,1,\cdots, M-1$, conforme tabela abaixo.

Mensagem	p_i
x_0	0.4
x_1	0.3
x_2	0.1
x_3	0.1
x_4	0.06
x_5	0.04

a) Determine um Código Ótimo $\boldsymbol{\theta}\{\cdot\}$ cujo conjunto de palavras-código $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ é formado a partir do alfabeto $\mathbf{A} = \{0,1\}$ (código binário).

b) Determine a Eficiência de $\boldsymbol{\theta}\{\cdot\}$.

c) Determine se $\boldsymbol{\theta}\{\cdot\}$ é Absolutamente Ótimo ou Quase Absolutamente Ótimo.

Solução:

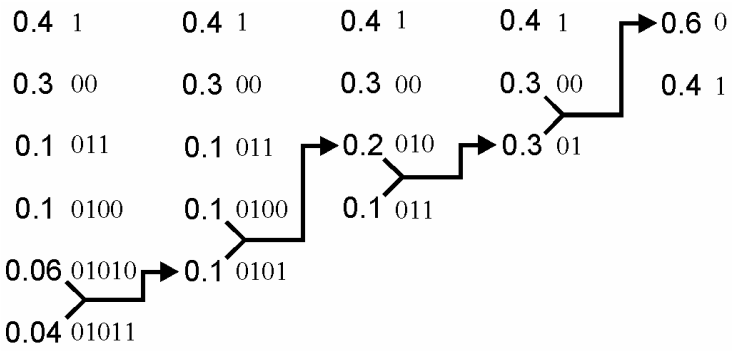

Figura 3.2: Procedimento para construção do Código de Huffman do Exemplo 3.13.

[F2] Comentário: Huffman1.pcx

Portanto, o código $\boldsymbol{\theta}\{\cdot\}$ resultante é

Mensagem	p_i	Palavra-código (símbolo) s_i associada a x_i por $s_i = \boldsymbol{\theta}\{x_i\}$
x_0	0.4	1
x_1	0.3	00
x_2	0.1	011
x_3	0.1	0100
x_4	0.06	01010
x_5	0.04	01011

$$H(X) = -\sum_{i=0}^{M-1} p_i \log_2(p_i) = 2.14 \ \ [bits/mensagem] \tag{3.12}$$

$$\overline{L}(\boldsymbol{\theta}) = \sum_{i=0}^{M-1} p_i \ell_i = \tag{3.13}$$
$$= 0.4\times 1 + 0.3\times 2 + 0.1\times 3 + 0.1\times 4 + 0.06\times 5 + 0.04\times 5 = 2.20 \ \ [bits/símbolo]$$

$$\eta = \frac{H(X)}{\overline{L}} = \frac{2.14 \ \ [bits/mensagem]}{2.20\,[bits/símbolo]} = 97.3\% \tag{3.14}$$

Visto que $H(X) \le \overline{L} < H(X)+1$, $\boldsymbol{\theta}\{\cdot\}$ é Quase Absolutamente Ótimo.

Exemplo 3.14: Seja uma fonte de informação representada pela variável aleatória discreta *X*, com espaço de amostras definido pelo conjunto $\mathbf{\Omega} = \{x_i\} = \{x_0, x_1, \cdots, x_{M-1}\}$ de $M = 2$ eventos estatisticamente independentes x_i com probabilidade de ocorrência p_i, $i = 0,1,\cdots, M-1$, conforme tabela abaixo.

Mensagem	p_i
x_0	0.6
x_1	0.4

Para reduzir o descasamento residual o sistema de codificação de fonte aplica o processo de extensão de fonte de ordem $J = 2$.

a) Determine um Código Ótimo $\boldsymbol{\theta}\{\cdot\}$ cujo conjunto de palavras-código $\mathbf{S} = \{s_i\} = \{s_0, s_1, \cdots, s_{M-1}\}$ é formado a partir do alfabeto $\mathbf{A} = \{0,1\}$ (código binário).

b) Determine a Eficiência do código obtido em a).

Solução:

A extensão da fonte de ordem $J = 2$ resulta no seguinte conjunto de "super-mensagens":

"Super – Mensagem" $x_i x_j$, $i, j = 0,1,\cdots, M-1$	p_{ij}
$x_0 x_0$	$0.6 \times 0.6 = 0.36$
$x_0 x_1$	$0.6 \times 0.4 = 0.24$
$x_1 x_0$	$0.4 \times 0.6 = 0.24$
$x_1 x_1$	$0.4 \times 0.4 = 0.16$

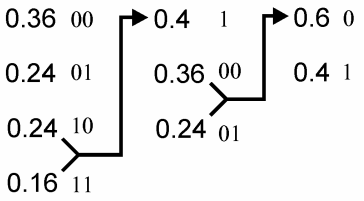

Figura 3.3: Procedimento para construção do Código de Huffman do Exemplo 3.14.

[F3] Comentário: Huffman2.pcx

Portanto, o código $\boldsymbol{\theta}\{\cdot\}$ binário resultante é

"Super-Mensagem" $x_i x_j$	p_{ij}	"Super-símbolo" s_{ij} associado a $x_i x_j$ por $s_{ij} = \boldsymbol{\theta}\{x_i x_j\}$
$x_0 x_0$	0.36	00
$x_0 x_1$	0.24	01

x_1x_0	0.24	10
x_1x_1	0.16	11

$$H(X) = -\sum_{i,j=0}^{M-1} p_{ij} \log_2\left(p_{ij}\right) = 1.94 \ \ [bits/mensagem] \tag{3.15}$$

$$\overline{L}(\boldsymbol{\theta}) = \sum_{i,j=0}^{M-1} p_{ij}\ell_{ij} = 0.36\times 2 + 0.24\times 2 + 0.24\times 2 + 0.16\times 2 = 2.00 \ \ [bits/símbolo] \tag{3.16}$$

$$\eta = \frac{H(X)}{\overline{L}} = \frac{1.94 \ \ [bits/mensagem]}{2.00\,[bits/símbolo]} = 97.0\% \tag{3.17}$$

3.3 O Algoritmo Lempel–Ziv

O Código de Huffman resulta em um código instantâneo que caracteriza-se por minimizar $\overline{L}$. Para construir um Código de Huffman adequado a uma fonte sem memória é necessário conhecer as probabilidades de ocorrência de cada mensagem.

Em contraste com o Código de Huffman, o algoritmo Lempel–Ziv é independente das estatísticas da fonte. Em função disto, o algoritmo Lempel–Ziv é enquadrado na classe de Algoritmos Universais para Codificação de Fonte. Neste estudo abordaremos apenas o caso em que a saída da fonte é uma seqüência de dígitos binários.

No algoritmo Lempel–Ziv a seqüência de bits proveniente da fonte é decomposta em blocos de tamanho variável denominados **frases**, as quais fazem o papel das mensagens só que de tamanho não fixo.

Uma nova frase é introduzida no conjunto $\mathbf{F} = \{\mathrm{f}_1, \mathrm{f}_2, \cdots, \mathrm{f}_N\}$ de N frases f_i, $i = 1, \cdots, N$, toda vez que um bloco de bits proveniente da fonte difere de alguma frase prévia já existente em $\mathbf{F}$ no último bit., quando, então, N é incrementado de 1. Esta frase prévia já existente em $\mathbf{F}$ que dá origem à nova frase é denominada de frase originadora, e é representada por $\mathrm{f}_o \in \mathbf{F}$. Assim, uma nova frase originada da respectiva $\mathrm{f}_o \in \mathbf{F}$ é idêntica à originadora exceto por possuir um bit adicional.

As frases assim formadas são listadas em um dicionário, o qual armazena a localização das frases existentes.

A codificação das frases em palavras-código consiste em especificar no campo de bits iniciais da palavra-código a localização (em base binária) da frase originadora e justapor a este campo de bits o último bit da nova frase.

Por exemplo, suponhamos que a fonte de informação emita a seqüência binária $x(n) = 10101101001001110101000011001110101100011011$.

O Algoritmo Lempel-Ziv decompõe $x(n)$ no conjunto $\mathbf{F}$ de frases obedecendo a regra que cada nova frase difere da respectiva $f_o \in \mathbf{F}$ no último bit: $\mathbf{F} = \{1, 0, 10, 11, 01, 00, 100, 111, 010, 1000, 011, 001, 110, 101, 10001, 1011\}$.

Observe que cada frase obtida de $x(n)$ é o resultado da concatenação da respectiva $f_o \in \mathbf{F}$ com um novo bit proveniente da fonte.

Para codificar as frases (isto é, para definir as palavras-código), o Algoritmo Lempel-Ziv constrói um dicionário conforme mostrado na Tabela 3.2.

Índice da Frase	Localização no Dicionário	**F**	Palavra-Código
1	0001	1	00001
2	0010	0	00000
3	0011	10	00010
4	0100	11	00011
5	0101	01	00101
6	0110	00	00100
7	0111	100	00110
8	1000	111	01001
9	1001	010	01010
10	1010	1000	01110
11	1011	011	01011
12	1100	001	01101
13	1101	110	01000
14	1110	101	00111
15	1111	10001	10101
16		1011	11101

Tabela 3.2: Dicionário resultante da seqüência $x(n)$ e formação das palavras-código.

As frases no dicionário são numeradas em ordem ascendente, começando com 1 até o número de frases resultantes da decomposição de $x(n)$, no caso, 16.

As palavras-código são determinadas listando no dicionário em base binária a localização da $f_o \in \mathbf{F}$ que origina a nova frase e justapondo a este campo de bits o último bit da nova frase. Inicialmente, a localização 0000 é utilizada para codificar frases que não apareceram anteriormente.

O decodificador no receptor digital constrói uma tabela idêntica à usada no codificador do transmissor a partir das palavras-código recebidas e recupera $x(n)$ lendo a coluna $\mathbf{F}$ de alto a baixo. Por exemplo, quando o receptor recebe a palavra-código 01010 o algoritmo consulta a localização 0101 e verifica que a frase originadora é 01. Portanto, justapondo o último bit da palavra-código à frase originadora obtemos a nova frase: 010. Obviamente, quando o receptor recebe a palavra-código 01010, a palavra-código 00101 correspondente à decodificação da frase 01 já foi recebida previamente, o que viabiliza a consulta recursiva ao dicionário.

É importante observar que a seqüência $x(n)$ proveniente da fone possui 44 bits e a codificação resultante da Tabela 3.2 gerou um conjunto de 16 palavras-código de 5 bits cada, o que implica em 80 bits enviados através do canal de transmissão. Então, o volume de bits transmitido foi AUMENTADO ao invés de diminuído! No entanto, esta ineficiência é devida ao fato de que $x(n)$ tem um tamanho muito pequeno. Este tamanho foi escolhido para os fins didáticos de possibilitar a compreensão do Algoritmo Lempel-Ziv. Na prática, em operação real, o algoritmo necessita, em geral, da ordem de 10^4 bits na seqüência $x(n)$ para que este tenha alguma eficiência. Obviamente seria difícil tornar didático um exemplo com 10^4 bits em $x(n)$!

Como o algoritmo seleciona o número N total de frases a serem armazenadas em $\mathbf{F}$? Em geral, não importando quão grande é o volume de memória que o sistema digital dispõe para armazenar $\mathbf{F}$, se nenhuma providência adicional for tomada, em algum momento do processo de codificação teremos problemas de *overflow* de memória. Para resolver este problema, o codificador e o decodificador de fonte devem possuir um algoritmo auxiliar que remove de $\mathbf{F}$ frases geradoras f_o em desuso e as substitui por novas que vão surgindo de acordo com o desenrolar temporal da seqüência de informação emitida pela fonte.

O Algoritmo Lempel-Ziv é largamente utilizado em aplicativos para compressão de arquivos como o Compress do UNIX , o PkZip do DOS e o Winzip do Windows.

3.4 Referências Bibliográficas

[Cover] T. M. Cover and J.A.Thomas , *Elements of Information Theory*, John Wiley & Sons, 1991.

[Ash] R. Ash, *Information Theory*, Interscience – John Wiley & Sons, 1967.

[Huffman] D.A.Huffman, "A Method for the Construction of Minimum Redundancy Codes", *Proceedings of IRE*, no. 40, pp. 1098-1101, 1952.