

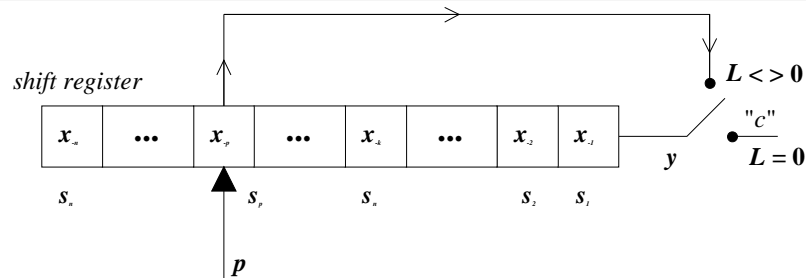


Sessão nº8

Codificação de Lempel-Ziv



Máquina para gerar seqüências de símbolos



Instruções possíveis para a máquina:

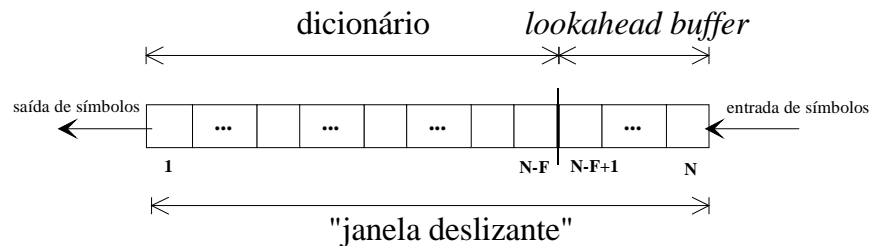
(L, p, c) se $L > 0$

$(0, c)$ se $L = 0$

- p informa a máquina onde posicionar o ponteiro
- L indica o número de clocks que a máquina deve receber com o ponteiro posicionado
- c é um elemento do alfabeto



LZ77 - Algoritmo da “janela deslizante”



- N é a dimensão da janela deslizante
- $N-F$ é a dimensão do dicionário que contem os símbolos já codificados
- F é a dimensão do *lookahead buffer* que contem os símbolos por codificar
- N e F são os parâmetros que caracterizam este algoritmo



Algoritmo LZ77 - a codificação



- Consideremos uma fonte de símbolos pertencentes ao alfabeto X que “alimenta” o *lookahead buffer*
- Pesquisar sobre os primeiros $N-F$ símbolos (dicionário) a maior subfrase contida na “janela deslizante” que coincide com a frase ou subfrase contida no *lookahead buffer*.
- Codificar o resultado da pesquisa na forma

$$(L, p, x)$$

- L é a dimensão da maior subfrase, $L \in [0, F]$
- p é a o ponteiro para a subfrase encontrada, $p \in [1, N-F]$
- x é a o primeiro símbolo do *lookahead buffer* que não coincide com a subfrase encontrada, $x \in X$
- Descodificação: é simples e rápida (não requer a pesquisa).



Factores determinantes nas variantes do LZ77



(compromisso entre a memória, o tempo e a eficiência de compressão)

- **tamanho do dicionário;** é fixo (ou não), ficando limitado (ou não) até que ponto da janela o ponteiro pode referenciar
 - dimensão em *bits* do ponteiro *versus* tempo e eficácia de compressão
- **tipo de *parsing*;** dentro dos limites da janela de texto considerada, determina qual das subfrases é referenciada pelo ponteiro
 - escolha da subfrase sem restrições de tempo e de memória (solução óptima)
 - escolha limitada a um conjunto de frases escolhidas de acordo com uma heurística (solução subóptima)
 - *greedy parsing*: mais comum e mais prático
 - *non-greedy parsing*: também permite bons resultados



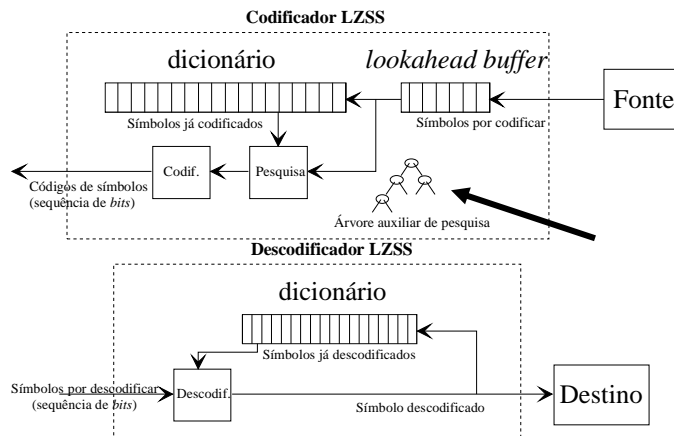
Estratégias de *parsing*: um problema



- **mensagem:** wooloomooloo
- **dicionário:** loo, loom, loomooloo, moo, ooloo, oom,oomoo, woo, wool
- **resultados possíveis do *parsing* da mensagem:**
 - woo-loo-moo-loo
 - woo-loom-ooloo
 - woo-loomooloo
 - wool-oom-ooloo
 - wool-oomoo-loo



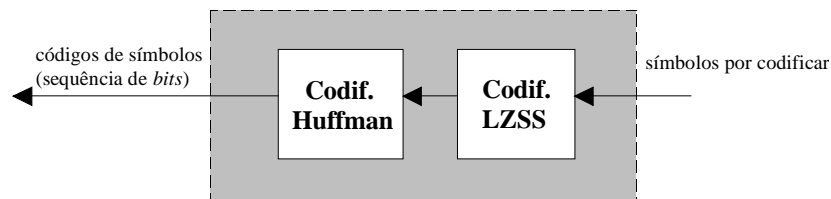
Estrutura do codificador/descodificador LZSS



- introdução da árvore auxiliar de pesquisa
 - palavras de códigos do tipo (L, p)
- número de *bits* $(L, p) = \log_2(F) + \log_2(N-F) + 1$



Diagrama de blocos do codificador LZHUF



- estrutura básica idêntica ao **codificador LZSS**
- utiliza-se **codificação de Huffman** na representação dos ponteiros p , da dimensão das subfrases L e dos símbolos x
 - modelo adaptativo de ordem 0
 - modelo estático de ordem -1, para os bits de maior peso dos ponteiros (tabela com valores pré-definidos ou tabela de *look up*)



Conjuntos de ficheiros para teste - exemplo



Nome do ficheiro	Dimensão [byte]
ETP94.ABS	186158
ETP94TA.ABS	186295
POSGASOL.ABS	193739
POSSHELL.ABS	181305
POSSIMPL.ABS	175260
POSVSHEL.ABS	199779

Nome do ficheiro	Dimensão [byte]
ETP94.HEX	530334
ETP94TA.HEX	530708
POSGASOL.HEX	552008
POSSHELL.HEX	516781
POSSIMPL.HEX	499533
POSVSHEL.HEX	569275

• Conjunto “ProgAbs” representativo de programas reais em ficheiros objecto no formato absoluto, para os terminais TP90 e TP92

• Conjunto “ProgHex” representativo de programas reais em ficheiros objecto no formato hexadecimal, para os terminais TP90 e TP92



Amostra dum ficheiro com programa no formato objecto absoluto hexadecimal da Intel



```
:02000002C0003C
:10000000FAB840008ED0BC0001B851008ED8BAA812
:10001000FFB8BCC0EFBAA6FFB8FC01EFBAA4FFB8A6
:100020003D00EFBA8001B80000EFBA0002B8E0006E
:10003000EF33C08EC08BF8268A0DF6D126880D26A8
:100040008A05C6066B5CFF2AC17405C6066B5C0098
:100050001E8EDFB900408B05F7D089053B05750D75

:100000008BD88BC2F7E18BF08BC7F7E303F08BC182
:05001000F7E303D6CB6D
:04000003FFFF0000FB
:00000001FF
```



Amostra dum ficheiro com programa no formato objecto absoluto da Intel



```
05 30 30 30 30 30 30 E4 84 1C 00 A8 C1 0B 43 4F
4E 54 41 20 41 20 43 52 45 44 49 54 41 52 20 20
20 20 20 20 20 20 A8 84 16 00 AA C1 03 20 41 43
54 49 56 41 52 20 50 4F 52 54 41 54 49 4C 20 1F
84 C8 01 D5 C0 0E 1E 0F D5 C0 7C 0F D5 C0 52 10
D5 C0 B0 10 D5 C0 44 11 D5 C0 1A 12 D5 C0 58 12
D5 C0 8A 12 D5 C0 4A 13 D5 C0 A8 13 D5 C0 FC 13
D5 C0 12 14 D5 C0 54 14 D5 C0 96 14 D5 C0 E8 14
D5 C0 58 15 D5 C0 E0 15 D5 C0 40 16 D5 C0 82 16
D5 C0 30 17 D5 C0 90 17 D5 C0 C2 17 D5 C0 7E 18
D5 C0 9A 18 D5 C0 52 19 D5 C0 62 1A D5 C0 C6 1A
D5 C0 82 1B D5 C0 2A 24 D5 C0 BE 24 D5 C0 D8 24
```



Conjunto normalizado para testes - Calgary corpus



Nome do ficheiro	Dimensão original [byte]
BIB	111261
BOOK1	768771
BOOK2	610856
GEO	102400
NEWS	377109
OBJ1	21504
OBJ2	246814
PAPER1	53161
PAPER2	82199
PIC	513216
PROGC	39611
PROGL	71646
PROGP	49379
TRANS	93695
Valor médio	224402

Este conjunto foi preparado na **Universidade de Calgary** (Canadá); pretende ser um conjunto de ficheiros normalizados para ser utilizado na investigação em compressão de dados.

É constituído por **10 ficheiros de texto**, 1 ficheiro com dados de geofísica, **2 ficheiros objecto** e 1 ficheiro com o *bitmap* duma imagem a preto e branco. Os de texto são constituídos por: 1 bibliografia, 1 livro de ficção e outro não de ficção, 1 registo de notícias, 2 artigos científicos, um programa escrito na linguagem “C”, 1 programa escrito na linguagem “Lisp”, 1 programa escrito na linguagem “Pascal” e 1 transcrição duma sessão num terminal.



Comparação de métodos de compressão - exemplo



- algoritmo de **Huffman** semi-adaptativo com modelo de ordem 0 (**HUFF**);
- algoritmo de **Huffman** adaptativo com modelo de ordem 0 (**AHUFF**);
- **codificação Aritmética** semi-adaptativa com modelo de ordem 0 (**ARITH**);
- **codificação Aritmética** adaptativo com modelo de ordem 1 (**ARITH1**);
- variante do algoritmo **LZ77** (Sliding Window) de Lempel-Ziv (**LZSS**);
- variante do algoritmo **LZSS** com codificação de **Huffman** estática (e modelo de ordem 0) da saída (**LZHUF**);
- variante do algoritmo **LZSS** com **codificação Aritmética** da saída (**LZARI**);
- variante do algoritmo **LZ78** de Lempel-Ziv (**LZW12**).



Programas de compressão* (didáticos)



Método	Descrição
HUFF	Algoritmo de Huffman com modelo semi-adaptativo de ordem 0
AHUFF	Algoritmo de Huffman com modelo adaptativo de ordem 0
ARITH	Codificação aritmética com modelo semi-adaptativo de ordem 0
ARITH1	Codificação aritmética com modelo adaptativo de ordem 1
LZSS	Codificação de Lempel-Ziv baseada no algoritmo LZ77, variante proposta por Storer e Szymanski em 1982, com os parâmetros $N=4096$ (número máximo de caracteres na janela deslizante) e $F=16$ (dimensão máxima da subfrase)
LZW12	Codificação de Lempel-Ziv baseada no algoritmo LZ78, variante proposta por Welch em 1984, com o parâmetro $M=4096$ (número máximo de frases no dicionário)

*implementações originais de Mark Nelson



Análise de recursos e resultados dos testes- exemplo



Método	Recursos para a descompressão		Resultados para o "ProgAbs"		Resultados para o "ProgHex"	
	Dimensão do código [byte]	Dimensão dos dados [byte]	Valor médio da eficácia de comp. [bits/byte]	Valor médio do tempo de descomp. [s]	Valor médio da eficácia de comp. [bits/byte]	Valor médio do tempo de descomp. [s]
HUFF	969	4365	6,92	2,00	3,87	2,17
AHUFF	1788	4664	6,80	2,50	3,86	3,33
ARITH	1049	682	6,89	3,33	3,83	9,17
ARITH1	997	136262	5,28	10,50	3,26	27,33
LZSS	302	4118	4,40	1,33	3,08	2,17
LZW12	380	30154	7,79	1,33	3,57	2,00
LZHUF	1225	9109	3,85	1,17	2,57	1,83
LZARI	1538	14918	3,84	1,83	2,56	3,33



Medidas de eficácia de compressão



- Relacionam a dimensão do texto com compressão β e a dimensão original do texto α .
- A **razão de compressão** é uma medida onde se consideram sempre blocos com 8 *bit* ou 1 *byte* do texto original, definida por

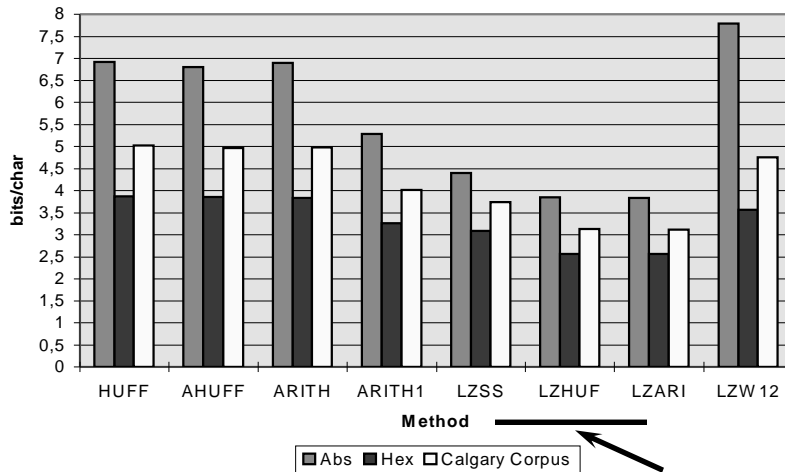
$$\text{razão de compressão} = 8 \times (\beta / \alpha) \text{ [bit/byte]}$$

- A **percentagem removida** é outra medida usada, definida por

$$\text{percentagem removida} = 100 \times (1 - \beta / \alpha) \text{ [%]}$$



Eficiência na compressão do Calgary Corpus



Recomendações para evitar a redundância



Com vista à maximização da entropia da fonte, nas mensagens a cifrar deve-se evitar:

- sinais de pontuação
- acentos
- mensagens do tipo “responda qual o número de mísseis?”
- a **mensagem** deve ser o **mais curto possível**

