



Teoria dell'Informazione e Applicazioni – a.a. 2014-2015

Esercizi su **Codifica** Lempel-Ziv 77-78

19-01-2015
Ing. P. Fazio

CODICI CHE DICTIONARY-BASED

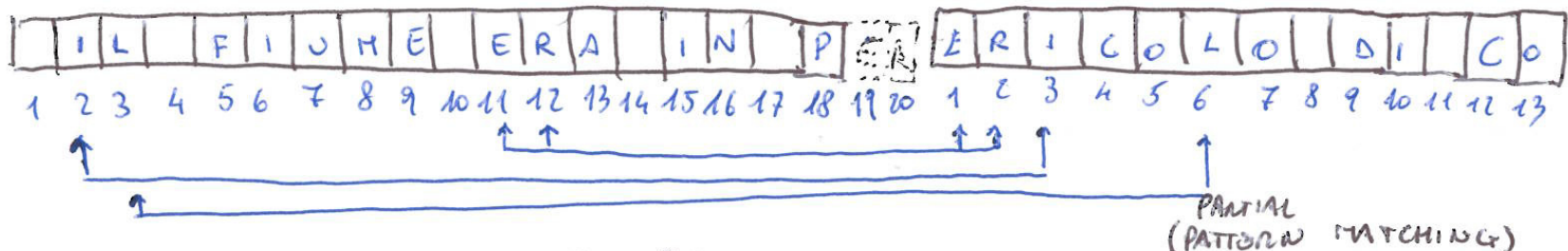
(27)

IN GENRALE, IN TUTTI I SISTEMI DI COMUNICAZIONE, LA POSIZIONE DI UNA LETTERA DELL'ALFABETO DIPENDE FORTEMENTE DALLA SINTASSI PASSATA (OVVERO DA QUALI LETTERE SONO STATE GIÀ USCITE PRECEDENTEMENTE); SI PARLA DI SORGENTI CON MEMORIA (L'ESEMPIO CLASSICO È IL TESTO SCRITTO).

L'IDEA È QUELLA DI USARE UN DIZIONARIO (VOCABOLARIO) CONTENENTE PAROLE A CUI SI ACCDE TRAMITE UN INDICE (IL DIZIONARIO È NOTO SIA AL COD. CHE AL DECOD.).

VOCABOLARIO (BUFFER CONTENENTE
CARATTERI GIÀ CODIFICATI)

CARATTERI DA CODIFICARE
(BUFFER CONTENENTE)



SI MOSTRA LA STRINGA COMUNE PIÙ LUNGA TRA BUFFER CONTENENTE E DIZIONARIO:
LE OCCORRENZE POSSONO ESSERE RIASSUNTE IN TABELLA COME MOSTRATO:

MATCH	INDICE INIZ. DIF.	LENGTH	INNOVAZIONE
1	11	2	//
1	2	1	//
0			"C"
0			"O"
1	3	1	//
...

L'IDEA È QUELLA DI RAPPRESENTARE I CARATTERI CON GLI INDICI TROVATI.

IN TAL MODO SI SFRUTTA IL RIPIETERSI DI ALCUNI PATTERN DI TESTO.

HUFFMAN È "ARITMETICA" SI BASA SUL CONCETTO DI PROBABILITÀ, CHE NON TIENE ASSOLUTAMENTE CONTO DELL'EFFETTO MORTO MA.

TRAMITE IL DIZIONARIO, INVECE, È POSSIBILE COSTRUIRE UN INSIEME DI SIMBOLI O SOSTITUIRE DI SIMBOLI DI USO FREQUENTE.

FORMALMENTE UN DIZIONARIO SI RAPPRESENTA CON LA TUPLA $D = (F, f)$, IN CUI F È L'INSIEME DI FRASI (SIMBOLI O SEQ. DI SIMBOLI) COSTRUITO SU UN ALFABETO A , f È LA FUNZIONE DI CODIFICA $f: F \rightarrow CW$, CW È L'INSIEME DELLE PAROLE CODIFICATE.

ESEMPIO:

$A = \{\alpha, \beta, \gamma, \delta\}$ $D = \{F, f\}$, CON $F = \{\alpha, \beta, \gamma, \delta, \alpha\delta, \alpha\delta, \beta\gamma\delta\}$ E

f È LA FUNZIONE CHE ASSOCIA I SIMBOLI ALLE PAROLE DI CODICE:

$f(\alpha) = 001$ $f(\alpha\delta) = 101$
 $f(\beta) = 010$ $f(\alpha\delta) = 110$
 $f(\gamma) = 011$ $f(\beta\gamma\delta) = 111$
 $f(\delta) = 100$

- DIZIONARI STATICI: SIMBOLI (E FRASI) E FUNZIONI SONO NOTI PRIMA DEL PROCESSO DI COMPILAZIONE E RIMANGONO INVARIATI

- DIZIONARI (ADATTATIVI) DINAMICI: È INIZIALMENTE VUOTO (O CONTIENE POCO SIMBOLI PRATICALI), SI AGGIORNA DINAMICAMENTE DURANTE IL PROCESSO

LETSCHEIDT'S PIÙ UTILIZZATO SONO QUELLI DI CAMPBELL-ZIV (FINE ANNI '70)

LZ77: COMPILAZIONE CON DIZIONARIO A FINESTRA SCORREVOLE (SLIDING WINDOW)
LA FINESTRA È DIVISA IN DUE PARTI: PARTE CONTENENTE SIMBOLI GIÀ CODIFICATI (SEARCH BUFFER) E PARTE CONTENENTE SIMBOLI DA CODIFICARE (LOOK-AHEAD BUFFER)

- STEP 1: SI SCEGLIE IL PRIMO CARATTERE DA CODIFICARE IN LAB;

- STEP 2: SI TROVA L'OCCORRENZA PIÙ LUNGA NEL SB (IL MATCH PUÒ ANCHE MOSTRARE NEL LAB)

- STEP 3: SI ANNOTA IL MATCHING TRAMITE UN PUNTATORE E SI SPosta IN AVANTI LA FINESTRA

- STEP 0: INIZIALMENTE SI CREA DEI PUNTATORI "VUOTI" (VEDUTO COME)

ES. 1: DATA LA STRINGA "cbbcc bcb bbb ccc c bcb", UTILIZZANDO LA CODIFICA LZ77 PER CODIFICARE LA STESSA, AVENDO $SIZE(SB) = 6$ E $SIZE(LAB) = 7$. INIZIALMENTE (POTREBBAMO $SB = \emptyset$, PERCUI:

c b b c c b c b b b b c c c c b c b

↑ SI ~~GUARDA~~ ^{PARCORRE} LA PRIMA "c" MA NON CI PUÒ ESSERE ALCUN MATCHING INQUANTO IL $SB = \emptyset$, QUINDI LA LUNGHEZZA DEL MATCHING È NULLA SI CREA IL PUNTATORE

$\langle 0, 0, f(c) \rangle$, IN CUI LA PRIMA CIFRA È L'OFFSET (LO SPOSTAMENTO)

AVANTI, LA SECONDA È LA LUNGHEZZA DI MATCHING E IL TERZO ELEMENTO È IL MAPPING ASSOCIATO. ^{RELATIVO AL SEMPLICE SUCCESSIVO AL LAB QUANDO C'È MATCH} SI SPOSTA A DX LA FINESTRA DI ^m POSIZIONI QUANTO SONO, IN CUI CON m UGUALE ALLA DIMENSIONE DEL MATCHING

SB LAB
c b b c c b c b b b b c c c c b c b ; NON TROVATE PER "b" NON HO MATCH, PER
WINDOW
c b b c c b c b b b b c c c c b c b ; NON TROVATE PER "b" NON HO MATCH, PER
c b b c c b c b b b b c c c c b c b ; NON TROVATE PER "b" NON HO MATCH, PER
c b b c c b c b b b b c c c c b c b ; NON TROVATE PER "b" NON HO MATCH, PER

OFFSET = 1 (SI DEVE USARE DI QUANTE POSIZIONI CI SI DEVE SPOSTARE A DX) (29)
LA LUNGHEZZA DEL MATCH È 1, PER CUI SI CREA $\langle 1, 1, f(c) \rangle$ E SI SPOSTA LA FINESTRA A DX DI 2 POSIZIONI (LUNGH. MATCH + 1): SI ALLUNGA IL SB E SI SPOSTA IL LAB

$\overbrace{c b b c}^{SD} \overbrace{c b c b b b b}^{LAB} c c c c b c b$ \downarrow dopo cb
 MATCH $\langle 4, 2, f(c) \rangle$, SI SPOSTA ADX DI 3 POS.

$c \overbrace{b b c c b c} \overbrace{b b b b c c c c} b c b$ \downarrow dopo bb
 MATCH $\langle 6, 2, f(b) \rangle$, SI SPOSTA ADX DI 3 POS.

$c b b c \overbrace{c b c b b b} \overbrace{b c c c c b c b}$ MATCH $\langle 5, 2, f(c) \rangle$, SI SPOSTA ADX DI 3 POS.

$c b b c c b c \overbrace{b b b b c c} c b c b$ MATCH $\langle 2, 2, f(b) \rangle$, SI SPOSTA ADX DI 3 POS.

$c b b c c b c b b b b \overbrace{b c c c c b c b}$ MATCH $\langle 2, 2, \text{NULL} \rangle$, END (LAB VUOTO)

QUINDI LA CODIFICA (COMPRESSIONS) SARÀ: $\langle 0, 0, f(c) \rangle \langle 0, 0, f(b) \rangle \langle 1, 1, f(c) \rangle$
 $\langle 4, 2, f(c) \rangle \langle 6, 2, f(b) \rangle \langle 5, 2, f(c) \rangle \langle 2, 2, f(b) \rangle \langle 2, 2, \text{EOF} \rangle$

DE CODIFICA (DATI I PUNTATORI, RICOSTRUIRE I SIMBOLI ORIGINALI):

$\langle 0, 0, f(c) \rangle \Rightarrow \text{STRINGA} = c$

$\langle 0, 0, f(b) \rangle \Rightarrow \text{STRINGA} = c b$ (accodato b)

$\langle 1, 1, f(c) \rangle \Rightarrow \text{STRINGA} = c b ; b c$

indicare il
 primo simbolo
 da ritruare a
 posizione iniziale

accodare b

$\langle 4, 2, f(e) \rangle \Rightarrow$ ebbe|ebc

$\langle 6, 2, f(b) \rangle \Rightarrow$ ebbe cbe|bbb

$\langle 5, 2, f(c) \rangle \Rightarrow$ ebbe cbebbb|bec

$\langle 2, 2, f(b) \rangle \Rightarrow$ ebbe cbe bbb bec|ecb

$\langle 2, 2, eof \rangle \Rightarrow$ ebbe ebe bbbb eee ebeb|END

mbit utilization = $8 \cdot \left[\overset{\substack{\text{modulation} \\ \text{max}}}{\text{mbit offset}}, \overset{\text{max}}{\text{mbit high. match max}}, \overset{\text{max}}{\text{mbit col/co}} \right] = 56$

ipotesi max " " "

3 3 1

mbit originari = $8 \cdot 18 = 144$

% guadagno = $\left(1 - \frac{\text{mbit ut}}{\text{mbit bcf.}} \right) \times 100 \approx 61\%$

ES. 2 : CODIFICARE SCONTO L277 LA STRINGA :

ABE AB AB ABED BBA

CON $\|SB\| = 4$
 $\|LAB\| = 6$

ABE AB AB ABED BBA $\Rightarrow \langle 0, 0, f(A) \rangle$ NO MATCH 1

ABE AB AB ABED BBA $\Rightarrow \langle 0, 0, f(B) \rangle$ " 2

ABE A BA B ABED BBA $\Rightarrow \langle 0, 0, f(e) \rangle$ " 3

ABE A BA B ABED BBA $\Rightarrow \langle 3, 2, f(A) \rangle$ MATCH, ^{AB} $f(A)$ 4

~~ABE AB AB ABED BBA~~

~~ABE~~ \downarrow PER NON SPAGLIARE CONVIENE SPOSTARE PRIMA IL LAB

A B E A B A BA BE D B B A $\Rightarrow \langle 2, 3, f(e) \rangle$ MATCH, ^{BAB} $f(e)$ 5

A B E A B A BA BE D B B A $\Rightarrow \langle 0, 0, f(D) \rangle$ NO MATCH 6

[illegible]
$$\text{max utility} = 8(2+2+2) = 48$$
$$\Rightarrow \% \text{ compression} = \left(1 - \frac{48}{112}\right) \times 100 \approx 57\%$$

8. 3. 2019
PUNJAB
IN PUNJAB CIRCULARS)

5 \Rightarrow A B C A B A : B A B C

OK

6 \Rightarrow A B C A B A B A B C D

4 \Rightarrow A B C A B A
 $\uparrow \uparrow \uparrow$
 1 2 3 $\underbrace{\hspace{1.5cm}}$
 4

$P \Rightarrow$ A B C A B A B A B A B C D B B

1 2 3 4 5 6 7

CODIFICA LZ78

31

- RISPETTO AL PRECEDENTE, SI ELIMINA IL CONCETTO DI FINESTRA (PER DIMENSIONI GRANDI DELLA FINESTRA SERVONO TROPPI BIT E AUMENTA LA COMPLESSITA' DEL PARSING)
- LA TUPLA E' COMPOSTA DA DUE ELEMENTI (NON PIU' 3):

\langle POS. NEL DIZ. DEL PRIMO SIMBOLO DELLA SEQ. DI MATCH / POS. NEL DIZ. DEL SIMBOLO SUCCESSIVO ALLA SEQ. DI MATCH \rangle

- TUTTE LE STRUNGHE DI SIMBOLI CODIFICATE ENTRANO NEL DIZIONARIO, FINO AD UNA DIM. MASSIMA (SE NON E' PIU' VALIDO LO SI AZZERAVA E SI RICOMINCIA).

ESEMPIO 1: SIA DATA LA STRINGA ACBBACBCCAA BBCCB

INDEX	POINTER	CODED SYMBOLS
1	$\langle 0, f(A) \rangle$	A
2	$\langle 0, f(C) \rangle$	C
3	$\langle 0, f(B) \rangle$	B
4	$\langle 3, 1 \rangle$	BA
5	$\langle 2, 3 \rangle$	CB
6	$\langle 2, 2 \rangle$	CC
7	$\langle 1, 1 \rangle$	AA
8	$\langle 0, 3 \rangle$	BB
9	$\langle 6, 3 \rangle$	CCB

SI PARTE DALLA LETTERA A, IL DIZIONARIO E' VUOTO

SI COSTRUISCE LA COPPIA $\langle 0, f(A) \rangle$ IN QUANTO NON C'E' IL SIMBOLO NEL DIZIONARIO, SENZA MATCHING. QUINDI IL CODICE LZ78 SARA' LA SEQ. DI P.TORI PRESENTI NELLA SECONDA COLONNA

PER LA DECODIFICA:

- | | |
|----------------------|-----------------------------------|
| 1 \Rightarrow A | 5 \Rightarrow ACBBACB |
| 2 \Rightarrow AC | 6 \Rightarrow ACBBACBCC |
| 3 \Rightarrow ACB | 7 \Rightarrow ACBBACBCCAA |
| 4 \Rightarrow ACBB | 8 \Rightarrow ACBBACBCCAA BB |
| | 9 \Rightarrow ACBBACBCCAA BBCCB |

ESSEMPIO 2) CODIFICARE CON LZ78 LA STRINGA "BACDBACCABDDDBABA"
BACDBACCABDDDBABA

INDEX	POINTSL	COORD. DATA
1	$\langle 0, f(B) \rangle$	B
2	$\langle 0, f(A) \rangle$	A
3	$\langle 0, f(C) \rangle$	C
4	$\langle 0, f(D) \rangle$	D
5	$\langle 1, 2 \rangle$	BA
6	$\langle 3, 3 \rangle$	CC
7	$\langle 2, 1 \rangle$	AB
8	$\langle 4, 4 \rangle$	DD
9	$\langle 4, 1 \rangle$	DB
10	$\langle 7, 2 \rangle$	ABA

B
 BA
 BAC
 BACD
 BACD BA
 BACD BA
 BACDBACC
 BACDBACCAB
 BACDBACCABDD
 BACDBACCABDDDBABA

ESSEMPIO 3:

DATA LA STRINGA "ABEBA BAC BAB BAC", COMPATIRLA CON LZ78 (32)

IND.	POINTSL	COORD. G.
1	$\langle 0, f(A) \rangle$	A
2	$\langle 0, f(B) \rangle$	B
3	$\langle 0, f(E) \rangle$	ABE
4	$\langle 2, 1 \rangle$	BA
5	$\langle 4, 3 \rangle$	BAC
6	$\langle 4, 2 \rangle$	BAB
7	$\langle 5, 3 \rangle$	BACC

A
 AB
 ABE
 ABEBA
 ABEBA BAC
 ABEBA BACBAB
 ABEBA BACBABACC OK



Possibili esercizi d'esame:

- **Codici a ripetizione** (ad esempio: data la stringa, codificare la stessa secondo un dato codice, valutare la rivelabilità/correggibilità degli errori; data una probabilità massima, trovare il codice a ripetizione che rispetti il vincolo);
- **Codici concatenati** (stesse valutazioni del caso precedente);
- Possibile confronto dei due casi precedenti;
- **Codici di Hamming** (ad esempio: determinazione matrice generatrice e di parità);
- Esercizi sul **calcolo probabilistico** e **teorema di Bayes**;
- **Codifica di Huffman** e procedura della lista concatenata (ad esempio: calcolo lunghezza media ed entropia);
- **Codifica aritmetica** (ad esempio: valutazione della precisione in base al numero di bit disponibili, conversione del tag da decimale a binario);
- **Codifica LZ77, LZ78.**