



UNIUNEA EUROPEANĂ



GUVERNUL ROMÂNIEI



Instrumente Structurale  
2007-2013



# Platformă de e-learning și curriculum e-content pentru învățământul superior tehnic

## Transmisia datelor multimedia in rețele de calculatoare

### 12. Compresia aritmetica cu scalare

# Compresia aritmetica cu scalare

- Problema implementării codării aritmetice este dată de precizia finită a operațiilor cu numere reale
- Soluția constă în înlocuirea intervalului  $[0,1)$  al numerelor reale cu intervalul  $[0 \dots 2^{N-1}]$  al numerelor întregi, unde  $N$  este un număr întreg, cu semnificație de numărul de cifre binare al registrelor numerice
- Ideea constă în selectarea prefixului comun la capetelor intervalelor,  $L$  și  $H$ , și apoi completarea capătului din stânga,  $L$ , cu simboluri 0 și al capătului dreapta,  $H$ , cu 1
- Nu este nevoie de un model al sursei, ci numai de alfabetul sursei
- Frecvențele simbolurilor sunt construite dinamic, deci se poate considera apartenența metodei la clasa metodelor dinamice (adaptive)

# Compresia aritmetica cu scalare

- Intervalele asociate simbolurilor alfabetului se calculeaza cu ajutorul frecventelor simbolurilor, in mod dinamic ; fiecare frecventa a unui caracter este initializata cu 1 si este incrementata de fiecare data cand apare simbolul in mesajul de comprimat
- Fie frecventa simbolului in alfabet
  - Frecventa cumulata a simbolurilor este

$$cum\_freq(i) = \sum_{j=i+1}^{N_s} freq(j)$$

unde  $N_s$  este numarul de simboluri din alfabetul sursei

- Rezulta ca  $cum\_freq(0)$  este frecventa cumulata a tuturor simbolurilor
- De retinut ca  $cum\_freq(0)-N_s-1$  este lungimea prefixului intrarii scanate
- Simbolurile sunt mentinute in ordinea descrescatoare a frecventelor lor

# Compresia aritmetica cu scalare

- Cand un simbol este citit din sursa text, intervalul curent  $[L, H)$  este ajustat la

$$L = L + \frac{(H - L + 1) \cdot cum\_freq(i)}{cum\_freq(0)}$$

$$H = L + \frac{(H - L + 1) \cdot cum\_freq(i - 1)}{cum\_freq(0)} - 1$$

- La fiecare ajustare a intervalului, se verifica conditia

$$H - L \geq cum\_freq(0)$$

# Compresia aritmetica cu scalare

- In caz contrar, intervalul este expandat cu relatia

$$[H, L) \rightarrow [2 \cdot (L - 2^{N-2}), 2 \cdot (H - 2^{N-2}) + 1)$$

- Efectuarea acestei transformari este memorata prin incrementarea unui numarator (*waiting\_counter*)
- Operatia se repeta atat timp cat intervalul este prea scurt
- Dupa salvarea mesajului comprimat, se trimite continutul numaratorului
- Dupa trimiterea ultimului bit al numaratorului, se transmite bitul invers de un numar de ori dat de continutul numaratorului de asteptare

# Algoritm

Codare\_simbol(ai)

```
L=L+(H-L+1)*cum_freq(i)/cum_freq(0) ;
H=L+(H-L+1)*cum_freq(i-1)/cum_freq(0)-1 ;
```

```
WC =0; // waiting_counter
```

**REPEAT**

**IF** (L and H au bit comun in stanga)

**THEN**

```
code = code + bit_comun;
```

```
L=L*2;
```

```
H=2*H+1;
```

**ELSE**

**IF** (H-L)<cum\_freq(0)

**THEN**

```
L = L*(L-2N-2);
```

```
H = 2*(H-2N-2) + 1;
```

```
Waiting_counter = ++1;
```

**UNTIL** (L and H nu au biti comuni in stanga) **and** (H-L) >= cum\_freq(0).

END.

Procedura de codare se incheie cu scrierea bitilor de asteptare (daca sunt) conform codului :

Write(bit,fout);

**WHILE** (waiting\_counter>0) **DO**

**IF** bit >0,

**THEN**

Write 1 in fout;

Write 0 in fout;

Waiting\_counter = --1;

End while;

END.