

# Structuri de date – Curs 13

Conf. univ. dr. Cristian CIUREA  
Departamentul de Informatica si Cibernetica Economica  
Academia de Studii Economice din Bucuresti  
[cristian.ciurea@ie.ase.ro](mailto:cristian.ciurea@ie.ase.ro)

# Agenda

- ▶ Compresia de date
- ▶ Algoritmi de compresie

# Compresia de date

- ▶ **Compresia** datelor este procedeul prin care se realizează reducerea spațiului ocupat pe suport de un fișier sau de un set de date. Prin compresie datele inițiale sunt transformate obținându-se reprezentări echivalente numite și date compresate.
- ▶ **Decompresia** este procedeul care asigură revenirea la forma inițială a unui fișier, adică datele compresate sunt aduse la o formă cât mai apropiată sau chiar identică cu forma pe care au avut-o înaintea compresiei.

# Compresia de date

Caracteristici:

- ▶ proces de codificare;
- ▶ utilizarea unui număr mai mic de biți pentru stocarea datelor;
- ▶ functionează dacă emițătorul și receptorul au algoritmul de codificare/decodificare;
- ▶ avantaj: reducerea gradului de utilizare a resurselor (HDD, lățime de bandă, etc.);
- ▶ dezavantaj: proces eventual costisitor pentru codificare/decodificare;
- ▶ algoritmi: fără/cu pierdere de informație.

# Compresia de date

- ▶ După momentul în care se face compresia, algoritmi sunt:
  - **algoritmi de compresie printr-o singură trecere sau algoritmi dinamici:** pe măsură ce se parcurge fișierul are loc și compresia;
  - **algoritmi de compresie prin mai multe treceri sau algoritmi statici:** la prima trecere se analizează alfabetul, frecvențele simbolurilor, se identifică subșirurile care se repetă, iar la următoarele treceri are loc particularizarea alfabetului destinație strict la particularitățile pe care le prezintă subalfabetul fișierului sursă.

# Compresia de date

Algoritmi dinamici:

- ▶ RLE (Run Length Encoding);
- ▶ LZW (Lempel–Ziv–Welch);
- ▶ etc.

# Compresia de date

Algoritmi statici:

- ▶ Huffman;
- ▶ Fano-Shannon;
- ▶ Compresia aritmetică;
- ▶ etc.

# Compresia de date

Algoritmi fără pierdere de informație:

- ▶ profită de redundanța statistică;
- ▶ date compresate fără erori;
- ▶ reversibili: datele sunt reconstituite în formatul original.



# Compresia de date

Algoritmi cu pierdere de informație:

- ▶ acceptă pierderea de conținut la codificare/decodificare;
- ▶ utilizați în funcție de modul de percepție a datelor;
- ▶ acceptare pierderi dacă rata de compresie este foarte ridicată.

# Compresia de date

Exemple algoritmi fără pierdere de informație:

- ▶ RLE;
- ▶ LZ;
- ▶ LZW;
- ▶ Huffman;
- ▶ etc.

# Compresia de date

Exemple algoritmi cu pierdere de informație:

- ▶ DCT: Discrete Cosine Transform;
- ▶ Compresie cu fractali;
- ▶ etc.

# Compresia de date

## RLE: Run-Length Encoding

- ▶ Secvente cu valori consecutive;
- ▶ Înlocuire secvență cu (frecvență apariție, valoare);
- ▶ aplicativitate: imagini cu repetiție mare a valorilor de reprezentare a culorilor.

Exemplu:

AAAAAAAAAANNAANNNNNNNN

10A2N5A7N

# Compresia de date

## LZ: Lempel–Ziv

- ▶ bazat pe lungimea codurilor identificate;
- ▶ construire dicționar cu grupuri de simboluri din datele compresate;
- ▶ pași algoritm:
  - inițializare dicționar cu blocurile de lungime 1;
  - căutarea celui mai mare (lungime) bloc care apare în dicționar;
  - codificare bloc cu index din dicționar;
  - adăugare în dicționar bloc concatenat cu primul simbol din blocul următor;
  - reluare pasul 2.

# Compresia de date

LZ: Lempel–Ziv

Exemplu:

A	N	N	A	A	B	B	A	A	B	A	B	B	A	A	A	A	B	A	A	B	B	A
0	1	1	0	2		4		2		6			5		5		7			3		0

# Compresia de date

## LZW: Lempel–Ziv–Welch

- ▶ îmbunătățire algoritm LZ;
- ▶ dicționar inițializat cu caracterele textului (o singură apariție);
- ▶ scanare șir intrare pentru subșiruri din ce în ce mai lungi până când este identificat unul care nu se află în dicționar;
- ▶ noul subșir, mai puțin ultimul caracter, este introdus în secvența codificată;
- ▶ noul subșir este adăugat în dicționar cu primul cod disponibil.

# Compresia de date

## Codul Huffman

- ▶ trebuie să se cunoască frecvența de apariție a caracterelor;
- ▶ pentru fiecare caracter se asociază o secvență de biți;
- ▶ secvența de biți construită pe baza unui arbore binar;



# Compresia de date

Algoritm Huffman:

- ▶ ordonare descrescătoare simboluri text compresat; criteriu: frecvența de apariție;
- ▶ un simbol reprezintă un nod în arbore; fiecare nod are asociată o frecvență de apariție;
- ▶ două noduri sunt legate dacă au asociate cele mai mici frecvențe de apariție; nodul părinte are asociată suma frecvențelor nodurilor legate;
- ▶ oprire algoritm: există un singur nod (nelegat).

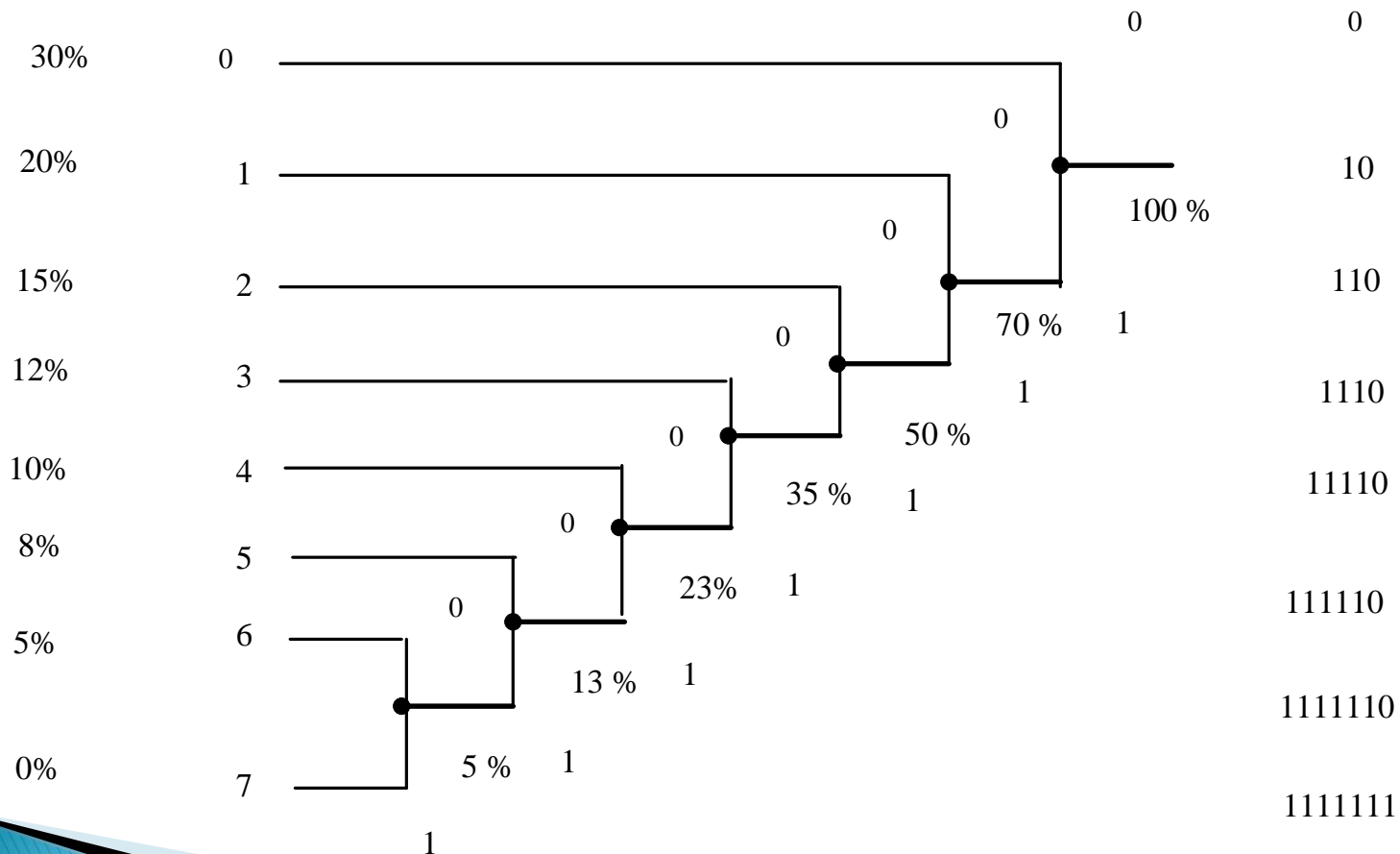
# Compresia de date

## ► Exemplu Huffman:

<b>Simbol</b>	<b>Nr. de apariții simbol</b>	<b>Total nr. de biți pentru un simbol (cod in clar)</b>	<b>Total nr. de biți pentru un simbol (cod Huffman)</b>
<b>0</b>	<b>12</b>	<b>96</b>	<b>12</b>
<b>1</b>	<b>8</b>	<b>64</b>	<b>16</b>
<b>2</b>	<b>6</b>	<b>48</b>	<b>18</b>
<b>3</b>	<b>5</b>	<b>40</b>	<b>20</b>
<b>4</b>	<b>4</b>	<b>32</b>	<b>20</b>
<b>5</b>	<b>3</b>	<b>24</b>	<b>18</b>
<b>6</b>	<b>2</b>	<b>16</b>	<b>14</b>
<b>7</b>	<b>0</b>	<b>0</b>	<b>0</b>
<b>Total</b>	<b>40</b>	<b>320</b>	<b>118</b>

# Compresia de date

## ► Exemplu Huffman:



# Compresia de date

## ▶ Exemplu Huffman:

- $S = [s1 ; s2 ; s3 ; s4 ; s5 ; s6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$

$s_2$       0.3

$s_1$       0.25

$s_3$       0.2

$s_4$       0.1

$s_5$       0.1

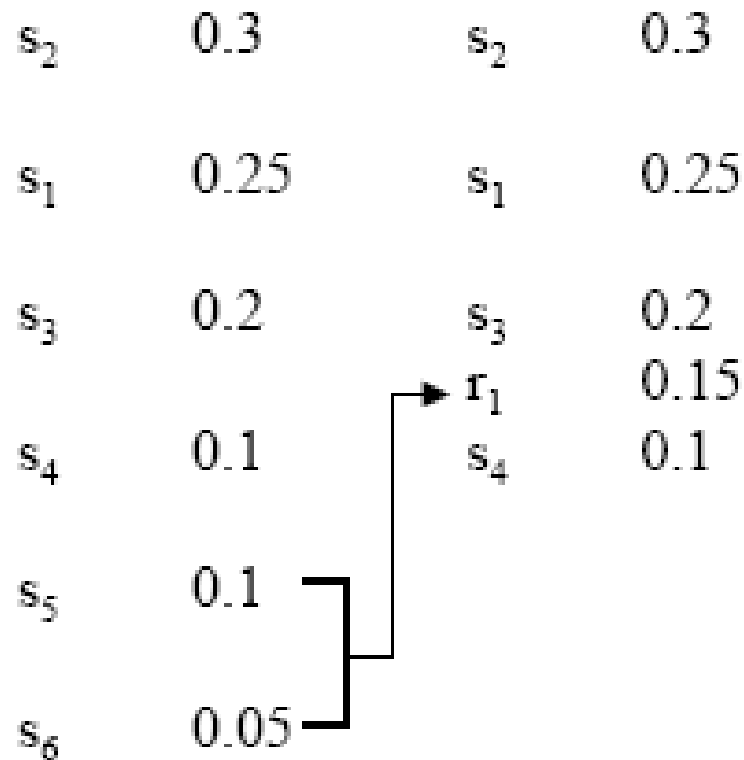
$s_6$       0.05

]

# Compresia de date

## ► Exemplu Huffman:

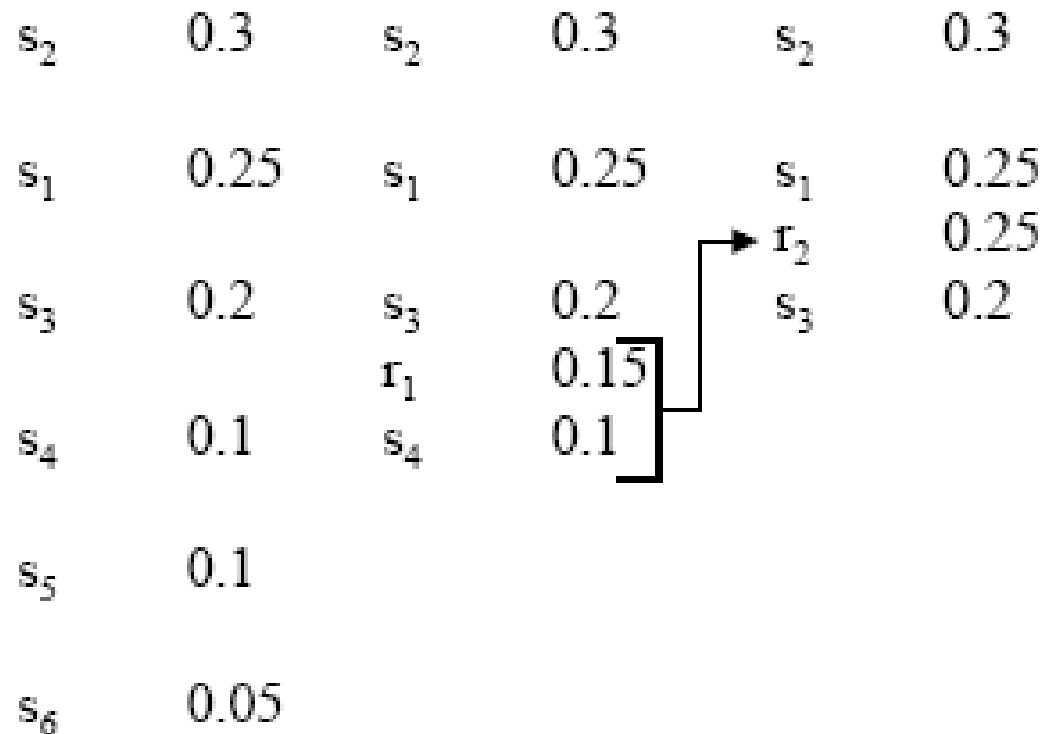
- $S = [s_1 ; s_2 ; s_3 ; s_4 ; s_5 ; s_6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$



# Compresia de date

## ► Exemplu Huffman:

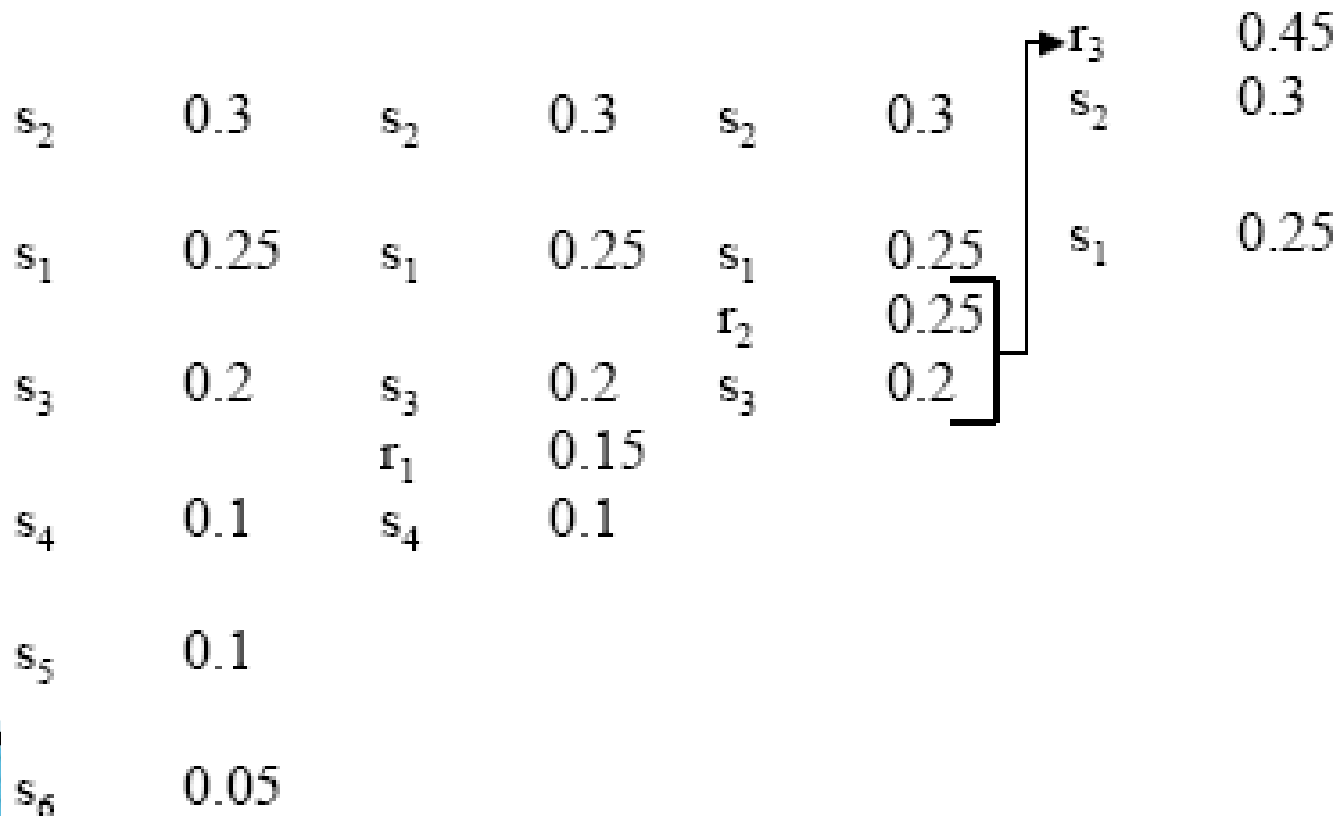
- $S = [s_1 ; s_2 ; s_3 ; s_4 ; s_5 ; s_6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$



# Compresia de date

## ► Exemplu Huffman:

- $S = [s1 ; s2 ; s3 ; s4 ; s5 ; s6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$



# Compresia de date

## ► Exemplu Huffman:

- $S = [s1 ; s2 ; s3 ; s4 ; s5 ; s6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$

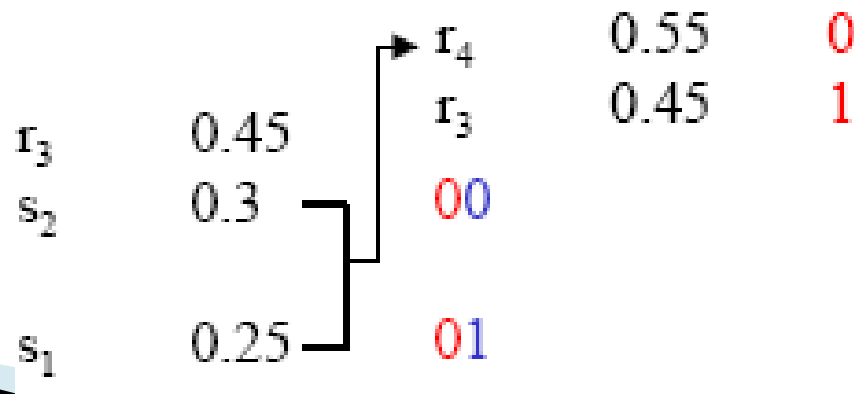
							$r_4$	0.55
							$r_3$	0.45
					$r_3$	0.45		
$s_2$	0.3	$s_2$	0.3	$s_2$	0.3	$s_2$	0.3	}
$s_1$	0.25	$s_1$	0.25	$s_1$	0.25	$s_1$	0.25	
				$r_2$	0.25			
$s_3$	0.2	$s_3$	0.2	$s_3$	0.2			
		$r_1$	0.15					
$s_4$	0.1	$s_4$	0.1					
$s_5$	0.1							
$s_6$	0.05							



# Compresia de date

## ► Exemplu Huffman:

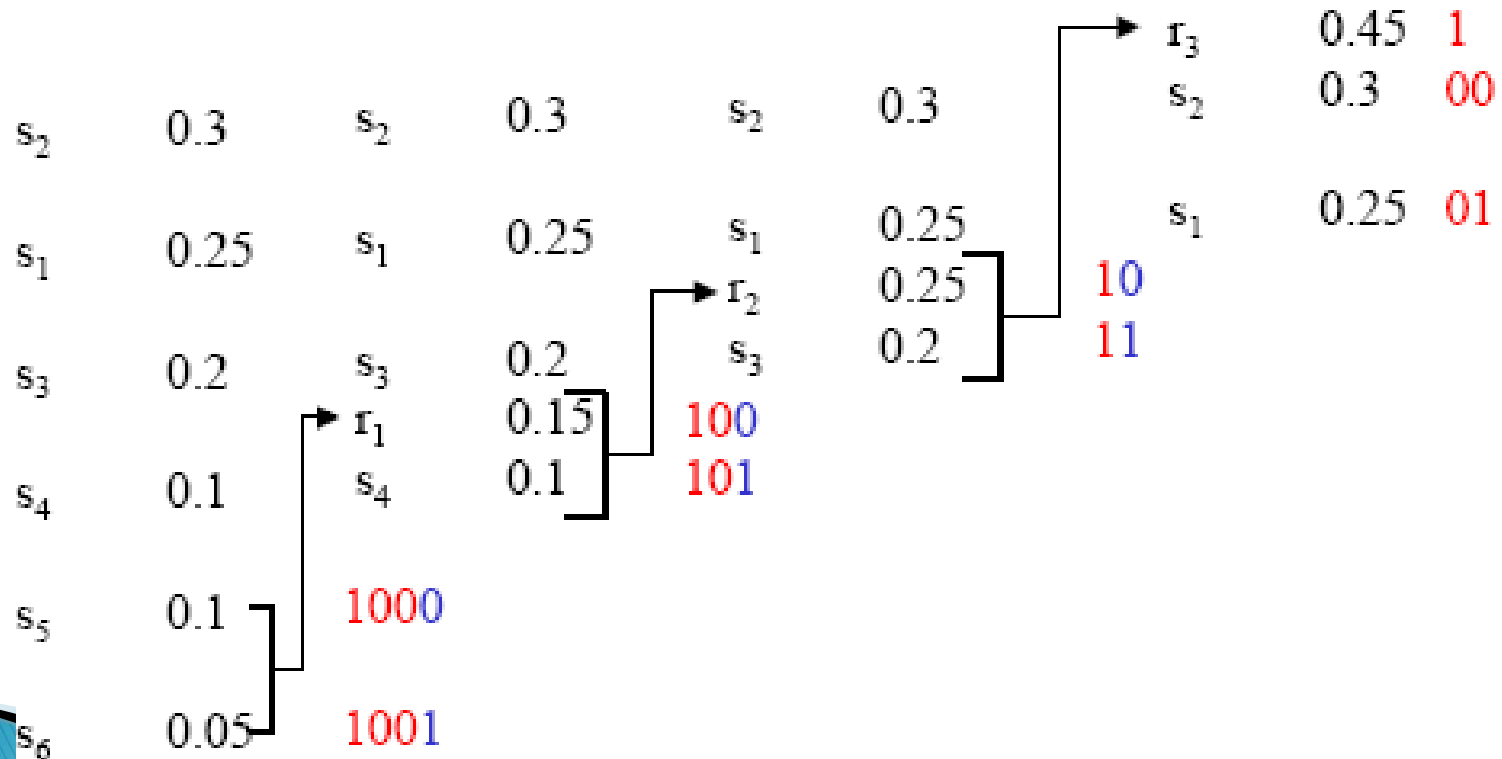
- $S = [s1 ; s2 ; s3 ; s4 ; s5 ; s6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$
- începând de la ultima sursă restrânsă se alocă câte un bit de 0 și 1 celor două simboluri; codurile simbolurilor reunite sunt prefix pentru codurilor simbolurilor ce le compun; se repetă alocarea de biți până când s-au codat toate simbolurile sursei inițiale.



# Compresia de date

## ► Exemplu Huffman:

- $S = [s1 ; s2 ; s3 ; s4 ; s5 ; s6]$
- $P = [0.25 ; 0.3 ; 0.2 ; 0.1 ; 0.1 ; 0.05]$



# Compresia de date

## ▶ Exemplu Huffman:

- se realizează tabelul de codare, care trebuie transmis împreună cu șirul de cuvinte de cod:

Simb.	Prob.	Cuvant cod	Lung. cod	Lung. medie a cuvintelor de cod :
$s_2$	0.3	00	2	$2 * 0.3 +$
$s_1$	0.25	01	2	$2 * 0.25 +$
$s_3$	0.2	11	2	$2 * 0.2 +$
$s_4$	0.1	101	3	$3 * 0.1 +$
$s_5$	0.1	1000	4	$4 * 0.1 +$
$s_6$	0.05	1001	4	$4 * 0.05 = 2.4 \text{ biti}$

# Bibliografie

- ▶ Ion Ivan, Marius Popa, Paul Pocatilu (coordonatori) – *Structuri de date*, Editura ASE, București, 2008.
  - Cap. 23. Compactarea și compresia datelor
  - Cap. 24. Utilizarea structurilor de date în compresia datelor