CAPITOLUL 3

COMPRESIA FĂRĂ PIERDERI PRIN CODARE HUFFMAN

La începutul acestui capitol sunt prezentate câteva preliminarii matematice privind compresia fără pierderi, necesare în analiza şi evaluarea procedurilor de codare din această categorie. În cazul transmisiilor la mică distanţă sau la puteri de emisie mari efectul perturbaţiilor este neglijabil, situaţii în care se pune numai problema realizării codării surselor discrete de informaţie de aşa manieră încât, pe de o parte, transmisia pe canal să poată fi posibilă, iar pe de altă parte, să se asigure un timp cât mai mic pentru transmiterea informaţiei sursei respective. În acest scop, sursa primară de informaţie cu alfabetul $S = \{s_1, s_2, ..., s_N\}$ este adaptată statistic la canalul de transmisiuni, care acceptă simbolurile din mulţimea $X = \{x_1, x_2, ..., x_M\}$, numită *alfabetul* de la intrarea canalului sau *alfabetul codului* folosit. Fiecărui mesaj s_k, $k = \overline{1, N}$, i se ataşează un cuvânt de cod, c_k, format dintr-o succesiune de simboluri $x_k \in X$, de aşa manieră încât timpul necesar transmiterii informaţiei sursei primare să fie minim.

3.1. Definirea codurilor nesingulare, unic decodabile şi instantanee

Fie $S=\{s_1,s_2,...,s_N\}$ mulțimea mesajelor unei surse discrete de informație, $X=\{x_1,x_2,...,x_M\}$ alfabetul codului şi $C=\{c_1,c_2,...,c_M\}$, mulțimea cuvintelor de cod. Prin operația de codare se realizează bijecția dintre mesajele $s_k\in S$ şi cuvintele de cod $c_k\in C$.

Prin definiție, un cod se numeşte *nesingular*, dacă toate cuvintele de cod sunt distincte.

Prin definiție, un cod se numeşte *unic decodabil*, dacă fiecărei succesiuni de simboluri recepționate îi corespunde o singură succesiune de mesaje ale sursei primare *S*.

Pentru fixarea ideilor, se consideră o sursă discretă de informație $S=\{s_1,s_2,s_3,s_4\}$ şi alfabetul codului $X=\{0,1\}$. Dacă mulțimea cuvintelor de cod este $C=\{c_1,c_2,c_3,c_4\}$, cu $c_1\rightarrow$l, $c_2\rightarrow$01, $c_3\rightarrow$10 şi $c_4\rightarrow$ l l, codul este nesingular, deoarece toate cuvintele de cod sunt distincte. Dacă, însă, se recepționează secvența 1101, aceasta poate fi interpretată în trei moduri diferite, şi anume fie s_4s_2, fie $s_1s_1s_2$, fie $s_1s_3s_1$, deci codul respectiv nu este unic decodabil.

O posibilitate de obținere a codurilor unic decodabile ar fi utilizarea unui simbol special în alfabetul codului, care să marcheze fie începutul fiecărui cuvânt de cod, fie sfârşitul acestuia. De exemplu, dacă simbolul α marchează sfârşitul fiecărui cuvânt de cod, atunci la recepționarea secvenței llα01α în condițiile exemplului considerat, se decide că s-a transmis succesiunea mesajelor s_4s_2, codul devenind astfel unic decodabil.

Datorită uşurinţei cu care se pot genera două stări, notate simbolic cu 0, respectiv 1, alfabetul codului este format numai din aceste două simboluri. De aceea, se pune problema de a se întocmi coduri unic decodabile folosind alfabetul binar $X = \{0,1\}$.

Considerând aceeaşi sursă discretă, ce poate furniza patru mesaje şi un alfabet de cod binar, se presupun următoarele două coduri codul *A*, în care $s_1 \to c_1 \to 0$, $s_2 \to c_2 \to 10$, $s_3 \to c_3 \to 110$, $s_4 \to c_4 \to 1110$ şi codul *B*, în care $s_1 \to c_1 \to 0$, $s_2 \to c_2 \to 01$, $s_3 \to c_3 \to 011$, $s_4 \to c_4 \to 0111$. Atât codul *A*, cât şi codul *B* sunt unic decodabile, deoarece în cazul codului *A* un zero marchează sfârşitul fiecărui cuvânt de cod, în timp ce în cazul codului *B* un zero marchează începutul fiecărui cuvânt de cod. Deşi ambele coduri sunt unic decodabile, între acestea există o diferenţă importantă în cazul codului *A*, decodarea (interpretarea) fiecărui cuvânt de cod se poate realiza odată cu recepţionarea integrală a acestuia, în timp ce în cazul codului *B* decodarea fiecărui cuvânt de cod trebuie să mai aştepte un timp, până la recepţionarea primului simbol din cuvântul următor.

Pentru a stabili diferenţa dintre cele două coduri, este necesar să se introducă noţiunea de prefix a unui cuvânt de cod.

Prin definiţie, succesiunea $x_{i_1} x_{i_2} \cdots x_{i_k}$, este *prefix* al cuvântului de cod $c_i \to x_{i_1} x_{i_2} \cdots x_{i_m}$, dacă $k \le m$.

Prin definiţie, un cod se numeşte *instantaneu*, dacă nici un cuvânt de cod nu este prefix pentru celelalte cuvinte de cod.

În cazul codului *B*, cuvântul de cod c_1, este prefix al cuvintelor c_2, c_3 şi c_4, cuvântul de cod c_2 este prefix al cuvintelor c_3 şi c_4, iar cuvântul de cod c_3 este prefix pentru c_4.

Evident, dacă un cod este instantaneu, el este şi unic decodabil, reciproca nefiind totdeauna adevărată.

Pentru a stabili dacă un cod este instantaneu, i se ataşează graful arborescent. În general, dacă alfabetul codului este $X = \{x_1, x_2, ..., x_M\}$, graful arborescent se întocmeşte astfel se pleacă dintr-un nod inițial, numit rădăcina grafului, care se desface în M ramuri (arce) pe care se alocă arbitrar cele M simboluri $x_1, x_2, ..., x_M$ din alfabetul codului. Din cele M noduri formate la capetele celor M ramuri se desfac din nou câte M ramuri, pe care se alocă arbitrar literele din alfabetul codului şi aşa mai departe.

Cuvintele de cod se formează citind ramurile, plecându-se de la rădăcina grafului arborescent spre nodurile terminale. Dacă toate cuvintele de cod corespund nodurilor terminale în graful arborescent, nici un cuvânt de cod nu este prefix pentru altul şi, deci, codul este instantaneu.

Grafurile arborescente pentru codurile *A* şi *B* sunt date în Fig. 3.1.

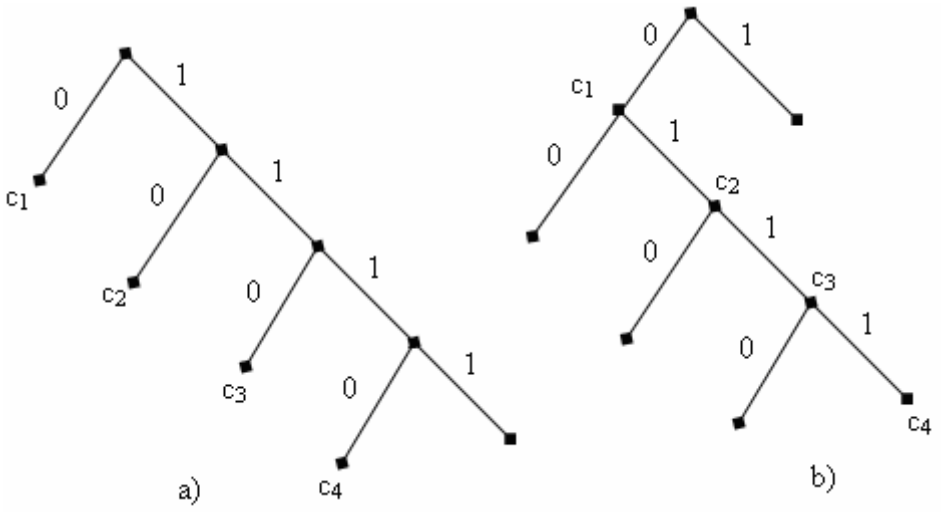


Fig. 3.1.a) Graful arborescent al codului instantaneu A; b) Graful arborescent al codului B

3.2. Teorema de existență a codurilor instantanee

Această teoremă stabileşte condiția necesară şi suficientă de *existență* a codurilor instantanee, referitoare la lungimile cuvintelor de cod.

Prin definiție, *lungimea* unui cuvânt de cod reprezintă numărul de simboluri din alfabetul codului, din care este format cuvântul respectiv.

Se consideră o sursă discretă de informație, caracterizată de mulțimea mesajelor $S = \{s_1, s_2, ..., s_N\}$, alfabetul codului $X = \{x_1, x_2, ..., x_M\}$, mulțimea cuvintelor de cod $C = \{c_1, c_2, ..., c_N\}$ şi mulțimea lungimilor cuvintelor de cod $L = \{l_1, l_2, ..., l_N\}$.

Un cuvânt de cod c_k este format dintr-o succesiune de l_k simboluri din alfabetul codului, de forma

$$c_k \rightarrow x_i ... x_j ... x_l$$

Condiția necesară şi suficientă *de existență* a codurilor instantanee este dată de inegalitatea lui Kraft, de forma

$$\sum_{k=1}^{N} M^{-l_k} \leq 1 \qquad (3.1)$$

unde M este numărul de simboluri din alfabetul codului, N este numărul cuvintelor de cod şi l_k - lungimea cuvintelor de cod.

Pentru a demonstra suficiența teoremei de existență, se consideră adevărată relația (3.1) şi, pe baza acesteia, se întocmeşte un cod care se va dovedi instantaneu.

Lungimile $l_1, l_2, ..., l_N$ ale cuvintelor de cod pot fi distincte sau nu. Fie n_1 numărul cuvintelor de cod formate numai dintr-un singur simbol (lungime egală cu unitatea), n_2 numărul cuvintelor de cod formate din două simboluri (lungime egală cu doi) şi aşa mai

departe, fie n_l, numărul cuvintelor de cod de lungimea cea mai mare, egală cu *l*.

Evident

$$\sum_{i=1}^{l} n_i = N \tag{3.2}$$

Cu aceste notații, rezultă că în inegalitatea (3.1) suma va conține n_1 termeni de forma M^{-1}, n_2 termeni de forma M^{-2} şi aşa mai departe, n_l termeni de forma M^{-l}. În consecință, inegalitatea (3.1) se poate scrie, echivalent, sub forma

$$\sum_{i=1}^{l} n_i M^{-i} \le 1 \tag{3.3}$$

Înmulțind relația (3.3) cu $M^l \ge 0$, rezultă

$$n_l \le M^l - n_1 M^{l-1} - \ldots - n_{l-2} M^2 - n_{l-1} M \tag{3.4}$$

Neglijând în relația (3.4) termenul $n_l \ge 1$ şi împărțind apoi noua relație cu $M \ge 2$, rezultă inegalitatea

$$n_{l-1} \le M^{l-1} - n_1 M^{l-2} - \ldots - n_{l-3} M^2 - n_{l-2} M \tag{3.5}$$

Neglijând în relația (3.5) termenul $n_{l-1} \ge 1$ şi împărțind apoi noua relație cu $M \ge 2$, rezultă inegalitatea

$$n_{l-2} \le M^{l-2} - n_1 M^{l-3} - \ldots - n_{l-3} M \tag{3.6}$$

În mod analog, se poate obține un şir de inegalități, ultimele trei fiind de forma

$$n_3 \le M^3 - n_1 M^2 - n_2 M \tag{3.7}$$

$$n_2 \le M^2 - n_1 M \tag{3.8}$$

$$n_1 \le M \tag{3.9}$$

Evident, dacă inegalitatea (3.1) este satisfăcută, cu atât mai mult vor fi satisfăcute inegalitățile (3.4) pană la (3.9).

Pe baza acestor inegalități se poate întocmi un cod

instantaneu, după cum urmează:

Se alege arbitrar un număr n_1 de cuvinte de lungime unu, care respectă inegalitatea (3.9). Rămân astfel disponibile $M - n_1$ prefixe formate dintr-un singur simbol, cu care se pot forma cuvinte de lungime egală cu doi, al căror număr este egal cu

$$\left(M - n_1\right)M = M^2 - n_1 M \tag{3.10}$$

Alegându-se arbitrar dintre acestea n_2 cuvinte ce respectă relația (3.8), rămâne disponibil un număr de prefixe din două simboluri, egal cu $M^2 - n_1 M - n_2$, cu care se pot forma cuvinte de lungime egală cu trei, în număr de

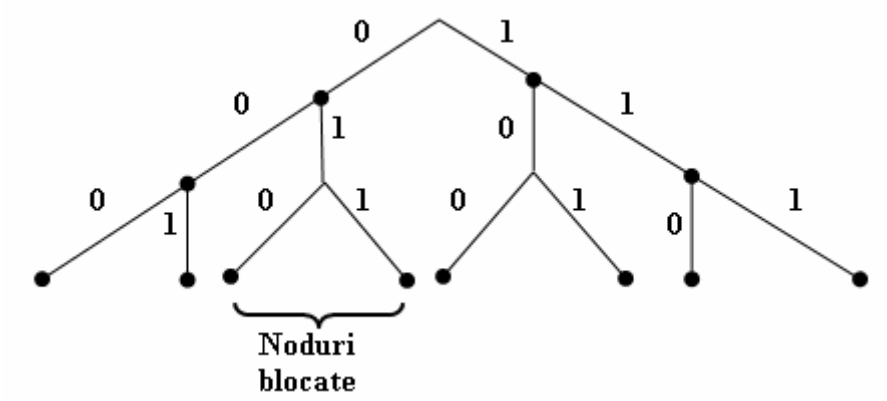
$$\left(M^2 - n_1 M - n_2\right)M = M^3 - n_1 M^2 - n_2 M \tag{3.11}$$

Dintre acestea, se aleg arbitrar n_3 cuvinte care respectă relația (3.7). În mod analog se formează şi celelalte cuvinte.

Deoarece nici un cuvânt nu devine prefix pentru altul, codul astfel întocmit este instantaneu, demonstrându-se astfel suficiența teoremei de existență a codurilor instantanee. Deoarece codurile instantanee formează o subclasă a codurilor unic decodabile, rezultă că s-a demonstrat suficiența teoremei de existență şi pentru codurile unic decodabile.

Pentru a demonstra necesitatea teoremei de existență a codurilor instantanee, se pleacă de la aserțiunea: cuvintele unui cod instantaneu pot fi aranjate totdeauna în nodurile terminale ale unui graf arborescent. Fie l cea mai mare lungime a cuvintelor de cod. La nivelul l arborele va avea, evident, un număr de M^l noduri. Fiecare cuvânt de lungime $l_k \le l$ blochează utilizarea a M^{l-l_k} noduri pe nivelul l. De exemplu, dacă l=3, M=2, (simbolurile "0" şi "1") şi se alege un cuvânt de lungime l_k = 2, fie acesta c_k = 01, atunci pe nivelul l = 3 se blochează utilizarea a $2^{3-2} = 2$ noduri, aşa cum este

arătat în figură care urmează.



Deoarece două cuvinte distincte blochează noduri diferite pe nivelul *l*, rezultă că numărul de noduri blocate pe nivelul *l*, rezultă că numărul de noduri blocate pe nivelul *l* este mărginit superior de numărul de noduri de pe acest nivel, adică

$$\sum_{k=1}^{N} M^{l-l_k} \le M^l \tag{3.12}$$

Simplificând cu $M^l > 0$, rezultă inegalitatea (3.1), care trebuia demonstrată.

Teorema *fiind de existență*, înseamnă că, dacă lungimile cuvintelor de cod satisfac inegalitatea (3.1), există cel puțin un procedeu de codare prin care să rezulte un cod instantaneu. Aceasta nu înseamnă că orice cod care satisface inegalitatea (3.1) este instantaneu. Teorema stabileşte că folosindu-se lungimi ale cuvintelor de cod ce satisfac inegalitatea (3.1), se poate întocmi cel puțin un cod instantaneu şi, deci, unic decodabil. Dacă pentru un cod dat este satisfăcută inegalitatea lui Kraft, pentru a decide dacă acesta este instantaneu, i se ataşează graful arborescent şi, dacă toate cuvintele de cod sunt noduri terminale, se decide că este

instantaneu.

Dacă pentru un cod dat inegalitatea (3.1) nu este satisfăcută, atunci, cu certitudine, codul nu este instantaneu şi nici unic decodabil şi, prin nici un procedeu de codare care va folosi aceleaşi lungimi ale cuvintelor de cod, nu se poate întocmi un cod instantaneu sau unic decodabil.

3.3. Lungimea medie a cuvintelor de cod

Pentru o sursă discretă de informaţie şi un alfabet de cod impus, se poate întocmi o mulţime de coduri instantanee sau unic decodabile. În scopul comparării acestora şi a alegerii celui sau celor mai bune (eficiente), se consideră drept criteriu de comparaţie timpul necesar transmiterii informaţiei sursei codate, de mărimea acestui timp fiind legate o serie de cheltuieli. Un cod instantaneu va fi cu atât mai eficient, cu cât timpul necesar transmiterii informaţiei sursei discrete va fi mai mic. Este posibil să se întocmească o multitudine de coduri instantanee de eficienţă maximă.

Pentru a compara diverse coduri instantanee, se va defini lungimea medie a cuvintelor de cod, care, aşa cum se va demonstra, este proporţională cu timpul mediu de transmitere a cuvintelor de cod. În felul acesta, se va demonstra că cel mai eficient cod instantaneu care se obţine, este cel pentru care lungimea medie a cuvintelor de cod este minimă.

Fie, pentru aceasta, o sursă discretă, completă şi fără memorie, caracterizată de distribuţia:

$$S:\begin{pmatrix} s_1 & \cdots & s_k & \cdots & s_N \\ p(s_1) & \cdots \mathrm{p}(s_k) & \cdots & p(s_N) \end{pmatrix} \tag{3.13}$$

Fie, de asemenea, mulţimea simbolurilor din alfabetul

codului
$X = \{x_1, x_2, ..., x_M\}$ şi mulţimea cuvintelor de cod $C = \{c_1, c_2, ..., c_N\}$.

Datorită corespondenţei bijective dintre $s_k \in S$ şi $c_k \in C$, rezultă

$$p(s_k) = p(c_k), \tag{3.14}$$

unde $p(c_k)$ reprezintă probabilitatea cuvântului c_k.

Informaţia ataşată mesajului s_k, notată cu $i(s_k)$, va fi atunci egală cu informaţia ataşată cuvântului c_k, notată cu $i(c_k)$, şi se va calcula cu
relaţia clasică

$$i(s_k) = i(c_k) = -\log p(s_k), \tag{3.15}$$

Dacă se notează cu $H(X)$ entropia codului utilizat, atunci, aceasta măsoară informaţia medie pe un simbol din alfabetul codului.

Fie *H(S)* entropia sursei ce urmează a fi codată, adică informaţia medie pe un mesaj.

Dacă $l_1, l_2, ..., l_N$ sunt lungimile cuvintelor de cod, lungimea medie a acestora se calculează cu relaţia

$$\bar{l} = \sum_{k=1}^{N} p(s_k) l_k \tag{3.16}$$

Pe de altă parte este evident că numărul mediu de simboluri ale unui cuvânt de cod, adică $\bar{l}$, este dat de raportul dintre informaţia medie pe cuvânt, *H(S),* şi informaţia medie pe un simbol constituent, adică

$$\bar{l} = \frac{H(S)}{H(X)} \tag{3.17}$$

Dacă se ţine cont de (1.12) şi (3.16), relaţia (3.17) se poate

scrie echivalent sub forma

$$\sum_{k=1}^{N} p(s_k) l_k = \frac{-\sum_{k=1}^{N} p(s_k) \log p(s_k)}{H(X)} \tag{3.18}$$

sau

$$\sum_{k=1}^{N} p(s_k) \left[l_k H(X) + \log p(s_k) \right] = 0 \tag{3.19}$$

O condiție suficientă, dar nu întotdeauna necesară, pentru satisfacerea relației (1.19) este ca

$$l_k = -\frac{\log p(s_k)}{H(X)} \tag{3.20}$$

Dacă se notează cu $\bar{\tau}$ durata medie de transmitere a unui simbol din alfabetul codului X, rezultă că durata de transmitere T_k a cuvântului de cod c_k, de lungime l_k, se poate determina cu relația

$$T_k = \bar{\tau} \cdot l_k \tag{3.21}$$

Duratele T_k, $k = \overline{1,N}$, de transmitere a cuvintelor de cod, determină o nouă variabilă aleatoare discretă, ce poate lua valori cu probabilitățile $p(s_k) = p(c_k)$ şi, deci, durata medie de transmitere a unui cuvânt de cod este

$$\bar{T} = \sum_{k=1}^{N} T_k p(s_k) = \bar{\tau} \cdot \sum_{k=1}^{N} p(s_k) l_k \tag{3.22}$$

Comparând relațiile (3.16) şi (3.22), rezultă

$$\bar{T} = \bar{\tau} \cdot \bar{l} \tag{3.23}$$

adică durata medie de transmitere a cuvintelor va fi cu atât mai mică, cu cât lungimea medie a acestora va fi mai mică.

Mărirea eficienței transmisiunii se poate, deci, obține selectând acel cod instantaneu care asigură o lungime medie minimă

a cuvintelor de cod.

Eficiența transmisiunii poate fi definită, dacă există o margine inferioară a lungimii medii, $\bar{l}$, a cuvintelor de cod.

Din relația (3.17) rezultă că micşorarea lui $\bar{l}$ se poate realiza fie prin micşorarea entropiei sursei ce urmează a fi codată, fie prin mărirea entropiei codului, adică mărirea informației medii pe un simbol din alfabetul codului.

Deoarece entropia sursei $H(S)$ nu poate fi modificată fără alterarea sursei inițiale, rezultă că singura posibilitate pentru micşorarea lungimii medii a cuvintelor de cod rămâne mărirea entropiei codului printr-o codare adecvată.

Notând cu $\bar{l}_{\min}$, lungimea medie minimă posibilă a cuvintelor de cod, se poate scrie relația

$$\bar{l}_{\min} = \frac{H(S)}{\max\left[H(X)\right]} \tag{3.24}$$

Deoarece în urma codării sursei *S*, utilizarea simbolurilor din alfabetul codului *X* depinde, în general, de unul sau mai multe simboluri folosite anterior, rezultă că, în general, sursa secundară *X* este o sursă cu memorie. La limită, când sursa secundară *X* devine fără memorie şi simbolurile acesteia sunt folosite echiprobabil, adică

$$p(x_1) = p(x_2) = \ldots = p(x_M) = \frac{1}{M}, \tag{3.25}$$

rezultă marginea superioară a entropiei sursei secundare X egală cu $\log M$ şi, deci, marginea inferioară a lungimii medii a cuvintelor de cod este $\dfrac{H(S)}{\log M}$.

În general, se poate scrie şirul de inegalități

$$\bar{l} \geq \bar{l}_{\min} \geq \frac{H(S)}{\log M} \tag{3.26}$$

În cazul particular, frecvent folosit în aplicații, al codurilor binare, M=2, caz în care relația (3.26) devine

$$\bar{l} \geq \bar{l}_{\min} \geq H(S) \tag{3.27}$$

Cu alte cuvinte, în cazul codurilor binare instantanee nu se pot obține lungimi medii ale cuvintelor de cod mai mici decât entropia sursei ce urmează a fi codată.

3.4. Eficiența şi redundanța unui cod

Prin definiție, se numeşte *eficiența* unui cod şi va fi notată cu η, raportul dintre marginea inferioară a lungimii medii a cuvintelor de cod şi lungimea medie a acestora, adică

$$\eta = \frac{\frac{H(S)}{\log M}}{\bar{l}} = \frac{H(S)}{\bar{l} \log M} \tag{3.28}$$

Înlocuind (3.20) în relația (3.28), rezultă o relație echivalentă de calcul pentru eficiența unui cod instantaneu, şi anume

$$\eta = \frac{H(X)}{\log M} \tag{3.29}$$

Prin definiție, se numeşte *redundanța* unui cod şi va fi notată cu ρ, mărimea complementară eficienței, adică

$$\rho = 1 - \eta \tag{3.30}$$

Din relația (3.29) rezultă că valoarea maximă a eficienței este $\eta = 1$.

Prin definiție, un cod se numeşte *absolut optimal*, dacă eficiența acestuia este maximă.

În cazul codului absolut optimal, din relația (3.28) rezultă

$$H(S) = \bar{l} \log M \tag{3.31}$$

Înlocuind în această relație $H(S)$ şi $\bar{l}$ cu relațiile lor de calcul, rezultă

$$\sum_{k=1}^{N} p(s_k)\left[l_k \log M + \log p(s_k) \right] = 0 \tag{3.32}$$

O condiție suficientă pentru satisfacerea acestei relații este

$$l_k \log M + \log p(s_k) = 0 \tag{3.33}$$

Din (3.33) rezultă

$$p(s_k) = M^{-l_k} \tag{3.34}$$

Sumând după k de la 1 la N în relația (3.34), se obține

$$\sum_{k=1}^{N} M^{-l_k} = \sum_{k=1}^{N} p(s_k) = 1, \tag{3.35}$$

adică în cazul codurilor absolut optimale inegalitatea lui Kraft devine egalitate. În acest caz mesajele trebuie furnizate cu probabilitățile date de relația (3.34).

3.5. Teorema codării surselor discrete, complete şi fără memorie pe canale neperturbate

S-a demonstrat în paragraful precedent că, dacă fiecare mesaj al unei surse discrete, complete şi fără memorie este furnizat cu probabilitatea data de relația (3.34), se poate obține un cod absolut optimal. Logaritmând în bază doi această relație, rezultă

$$l_k = -\frac{\log p(s_k)}{\log M} \tag{3.36}$$

Conform definiției lungimii cuvintelor de cod, rezultă că

numai în cazul codurilor absolut optimale raportul $-\frac{\log p(s_k)}{\log M}$ este un număr întreg pozitiv, ce poate fi considerat egal cu lungimea cuvântului de cod c_k.

În general, sursa discretă de informație își furnizează mesajele cu diverse probabilități care nu respectă relația (3.34) și, deci, raportul $-\frac{\log p(s_k)}{\log M}$ nu mai este un număr întreg. În acest caz, lungimile cuvintelor de cod, l_k , $k=\overline{1,N}$, se aleg numerele întregi pozitive care respectă dubla inegalitate

$$-\frac{\log p(s_k)}{\log M} \le l_k < -\frac{\log p(s_k)}{\log M} + 1 \qquad (3.37)$$

Alegând astfel lungimile cuvintelor de cod, se poate demonstra că inegalitatea iui Kraft este satisfăcută, deci, cu astfel de lungimi se poate întocmi totdeauna un cod instantaneu.

Într-adevăr, din prima inegalitate a relației (3.27) rezultă

$$M^{-l_k} \le p(s_k) \qquad (3.38)$$

Sumând după *k*, de la *k*=l la *k*=*N* și ținând cont că sursa discretă de informație este completă, rezultă

$$\sum_{k=1}^{N} M^{-l_k} \le \sum_{k=1}^{N} p(s_k) = 1 \qquad (3.39)$$

Multiplicând relația (3.37) cu $p(s_k) > 0$ și apoi sumând după k, de la k=1 la k=N, rezultă

$$\frac{-\sum_{k=1}^{N} p(s_k)\log p(s_k)}{\log M} \le \sum_{k=1}^{N} p(s_k)l_k < \frac{-\sum_{k=1}^{N} p(s_k)\log p(s_k)}{\log M} + \sum_{k=1}^{N} p(s_k) \tag{3.40}$$

Ţinând cont de (3.18), (3.19) şi de faptul că totdeauna sursele discrete de informaţie sunt complete, relaţia (3.40) devine

$$\frac{H(S)}{\log M} \le \bar{l} < \frac{H(S)}{\log M} + 1 \tag{3.41}$$

Relaţia (3.41) este adevărată pentru orice sursă S discretă, completă şi fără memorie, deci va fi adevărată şi pentru extensia sa de ordinul *m*.

Notând cu $H(S^m)$ entropia extensiei de ordinul *m* şi cu $\overline{l_m}$, lungimea medie a cuvintelor de cod corespunzătoare mesajelor compuse σ_k ale extensiei de ordinul *m*, relaţia (3.41) se poate scrie sub forma

$$\frac{H(S^m)}{\log M} \le \overline{l_m} < \frac{H(S^m)}{\log M} + 1 \tag{3.42}$$

Aşa cum s-a demonstrat în Capitolul 1,

$$H(S^m) = mH(S) \tag{3.43}$$

Ţinând cont de (3.43), relaţia (3.42) devine

$$\frac{H(S)}{\log M} \le \frac{\overline{l_m}}{m} < \frac{H(S)}{\log M} + \frac{l}{m} \tag{3.44}$$

Pe de altă parte

$$\frac{\overline{l_m}}{m} = \bar{l} \tag{3.45}$$

deoarece un mesaj compus este format dintr-o succesiune de *m* mesaje ale sursei inițiale *S*.

Din (3.44) şi (3.45) rezultă că, la limită, când $m \to \infty$, lungimea medie a cuvintelor de cod atinge marginea inferioară a lungimii medii a cuvintelor de cod, adică

$$\bar{l} = \frac{H(S)}{\log M} \tag{3.46}$$

obținându-se astfel un cod absolut optimal.

Conform acestei teoreme, rezultă că, efectuându-se codări pe extensii ale unei surse discrete de informație, se pot obține lungimi medii ale cuvintelor de cod cu atât mai mici, cu cât ordinul extensiei este mai mare. La limită, când ordinul extensiei tinde la infinit, se obține lungimea medie cea mai mică posibilă a cuvintelor de cod, egală cu marginea inferioară a acestora.

În practică nu se poate coda extensia de ordinul *m* a sursei primare, când *m* tinde la infinit. Ordinul extensiei va avea totdeauna o valoare finită şi, cu cât ordinul extensiei este mai mare, cu atât complexitatea instalației de codare creşte. Mai mult, de la o anumită valoare a ordinului extensiei, creşterea ordinului în continuare va duce la scăderi nesemnificative ale lungimii medii a cuvintelor de cod, astfel că nu se mai justifică prețul de cost ridicat, dictat de creşterea complexității instalației de codare. Din această cauză, în practică, în multe situații reale, se realizează codări pe mesaje individuale, utilizându-se procedee de codare care conduc la coduri instantanee de eficiență cât mai ridicată, adică acelea care conduc spre lungimi medii ale cuvintelor de cod cele mai mici posibile. Dintre procedeele de codare posibile s-au impus, procedeul de codare Huffman, procedeul de codare binară Shannon – Fano şi codarea aritmetică.

3.6. Procedeu de codare binară Huffman statică

Acest procedeu se bazează pe ideea de a partiționa mulțimea mesajelor sursei $S=\{s_1,s_2,...,s_N\}$ în submulțimile S_0 şi S_1, astfel încât suma probabilităților mesajelor incluse în S_0 să fie cât mai apropiată de suma probabilităților mesajelor incluse în S_1. La rândul lor, submulțimile S_0 şi S_1 pot fi partiționate în submulțimile S_{00} şi S_{01}, respectiv S_{10} şi S_{11} astfel încât suma probabilităților mesajelor incluse în cele patru submulțimi să fie cât mai apropiate posibil. Procedeul se continuă în mod similar până când se obțin submulțimi ce conțin un singur mesaj.

În felul acesta, pentru orice distribuție a sursei *S* ce urmează a fi codată se va obține un cod compact, adică lungimi medii ale cuvintelor de cod ce nu mai pot fi micşorate prin nici un alt procedeu de codare.

Pentru ca partițiile să satisfacă condițiile menționate, se procedează astfel:

1) Se ordonează mulțimea mesajelor sursei S în ordinea descrescătoare a probabilităților, obținându-se astfel mulțimea ordonată $R_0=\{s_1,s_2,...,s_N\}$, cu $p(s_1)\geq p(s_2)\geq ... \geq p(s_N)$, cu schimbarea eventuală a indicilor mesajelor pentru realizarea ordonării respective;

2) Se reunesc ultimele două mesaje (de probabilitățile cele mai mici) într-un nou mesaj, notat cu r_1, căruia i se alocă o probabilitate egală cu suma probabilităților mesajelor componente. Se ordonează din nou mesajele în ordinea descrescătoare a probabilităților, formându-se astfel prima sursă restrânsă $R_1=\{s_1,s_2,...,r_1,...\}$, cu $p(s_1)\geq p(s_2)\geq ... \geq p(r_1)\geq ....$

3) Se reunesc ultimele două mesaje din sursa restrânsă R_1 într-un nou mesaj r_2, de probabilitate egală cu suma probabilităţilor mesajelor componente. Se ordonează mesajele în ordine descrescătoare, formându-se astfel sursa restrânsă R_2. În mod analog, din R_2 se formează sursa restrânsă R_3 şi aşa mai departe, până când se obţine o sursă restrânsă formată numai din două mesaje, $R_n = \{r_n, r_{n-1}\}$, cu $p(r_n) \geq p(r_{n-1})$. De fapt, r_n va fi S_0 şi r_{n-1} va fi S_1 sau invers.

Din modul de formare a surselor restrânse R_i, rezultă că mulţimea S a mesajelor poate fi partiţionată în două submulţimi r_n şi r_{n-1} astfel încât probabilităţile $p(r_n)$ şi $p(r_{n-1})$ sunt cele mai apropiate posibil. La rândul lor, submulţimile r_n şi r_{n-1}, pot fi partiţionate în alte două submulţimi, de probabilităţile cele mai apropiate posibil. Partiţionările se continuă până se obţin submulţimi care conţin un singur mesaj.

4) Cuvintele de cod corespunzătoare fiecărui mesaj se obţin astfel:

- submulţimii r_n i se alocă simbolul "0" (sau "1");
- submulţimii r_{n-1}, i se alocă simbolul "1" (sau "0");
- la fiecare partiţionare se alocă arbitrar celor două submulţimi "0" sau "1", operaţia continuându-se până se obţin submulţimi ce conţin un singur mesaj s_k, $k = \overline{1, N}$.

Deoarece alocarea lui "0" şi "1" este arbitrară la fiecare partiţionare, rezultă că unei surse S i se pot ataşa o multitudine de coduri instantanee, toate, însă, având aceeaşi lungime medie a cuvintelor de cod, care nu mai poate fi micşorată prin nici un alt procedeu de codare a mesajelor luate individual.

Dacă sursa primară S poate furniza N mesaje, atunci submulțimea restrânsă R_1, va avea N-1 mesaje, submulțimea restrânsă R_2 va conține N-2 mesaje şi aşa mai departe, ultima submulțime restrânsă R_n va conține $N-n$ mesaje, care sunt r_n şi r_{n-1}, adică se poate scrie:

$$N-n=2 \Rightarrow n=N-2 \tag{3.47}$$

Dacă submulțimii r_n i se alocă simbolul "0" şi submulțimii r_{n-1} simbolul "1", celor N-2 partiționări putându-li-se aloca arbitrar "0" sau "1", rezultă un total de 2^{N-2} posibilități de codare. Dacă, însă, submulțimii r_n i se alocă simbolul "1", iar submulțimii r_{n-1} simbolul "0", mai rezultă 2^{N-2} posibilități de codare. Rezultă, deci, că prin acest procedeu de codare se pot realiza $2^{N-2}+2^{N-2}=2^{N-1}$ coduri instantanee, toate având toate aceeaşi lungime medie a cuvintelor de cod.

Prin definiție, se numeşte cod compact, codul care realizează lungimea medie minimă a cuvintelor de cod. Deoarece prin procedeul de codare Huffman se obține cea mai mică lungime medie a cuvintelor de cod, înseamnă că prin acest procedeu se obțin coduri instantanee compacte. Evident, un cod absolut optimal este şi compact, reciproca nefiind totdeauna valabilă.

Exemplul 3.1.

Se presupune sursa discretă de informație caracterizată de distribuția:

$$S:\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ 0,1 & 0,2 & 0,3 & 0,15 & 0,05 & 0,2 \end{pmatrix}.$$

Codarea binară Huffman a acestei surse se poate realiza astfel:

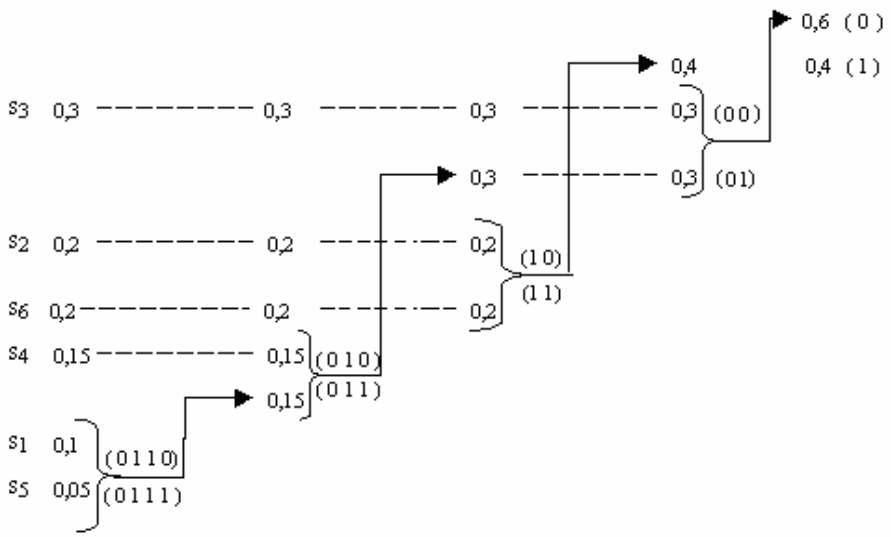


Fig. 3.2. Schema de codare pentru exemplul 3.1

Graful şi cuvintele de cod corespunzătoare codării efectuate sunt date în Fig. 3.3.

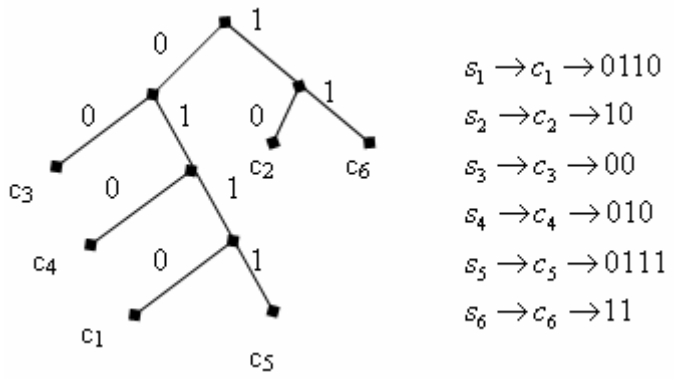


Fig. 3.3. Graful corespunzător codului

3.6.1. Coduri Huffman de dispersie minimă

Codurile Huffman de dispersie minimă se obțin când la

reordonarea sursei restrânse, simbolul compus se plasează pe poziția cea mai de sus posibil în sursa restrânsă. În felul acesta cuvântul de cod atribuit simbolului compus va avea cea mai mică lungime posibilă. Cum acest cuvânt va deveni prefix pentru simbolurile constituente, cuvintele de cod corespunzătoare acestora vor avea o lungime cu o unitate mai mare decât lungimea prefixului, deci şi acestea vor rezulta de lungime minimă. Ca urmare, diferențele dintre lungimile cuvintelor de cod devin minime, ceea ce va conduce, evident, şi la o dispersie minimă.

Pentru fixarea ideilor, se presupune sursa discretă de informație caracterizată de distribuția:

$$S:\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ 0{,}2 & 0{,}4 & 0{,}2 & 0{,}1 & 0{,}1 \end{pmatrix}$$

Pentru această sursă se efectuează codarea Huffman, plasând întâi mesajele sursei restrânse pe pozițiile cele mai jos posibile în listă şi apoi pe pozițiile cele mai de sus posibile.

În primul caz rezultă schema de codare din Fig. 3.4, iar graful şi cuvintele de cod ca în Fig. 3.5.

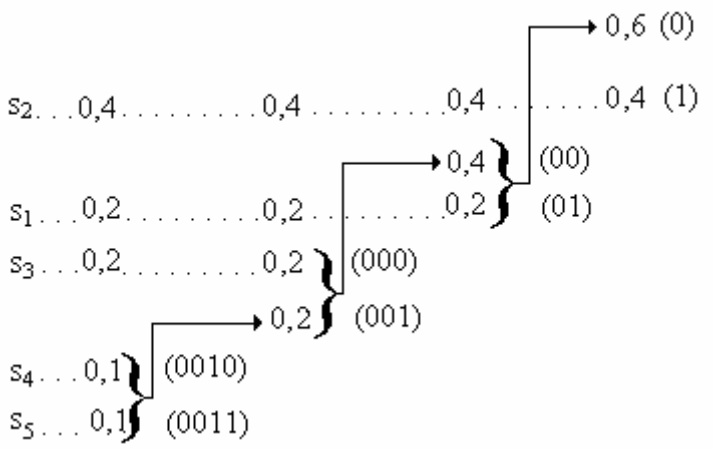


Fig. 3.4. Schema de codare

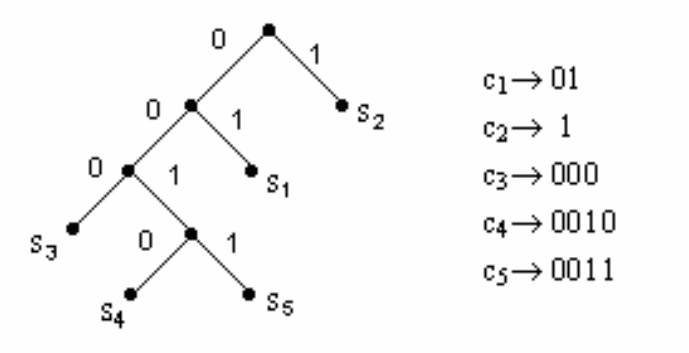


Fig. 3.5. Graful şi cuvintele de cod

Pentru acest cod, lungimea medie şi dispersia sunt:

$\overline{l} = 0,2 \cdot 2 + 0,4 \cdot 1 + 0,2 \cdot 3 + 0,1 \cdot 3 + 0,1 \cdot 4 = 2,2$ biţi/mesaj

$$\sigma_1^2 = \sum_{i=1}^{5} \left(l_i - \overline{l} \right)^2 = 0,2^2 + 1,2^2 + 0,8^2 + 1,8^2 + 1,8^2 = 8,6$$

Pentru cazul în care în codarea Huffman mesajele sursei restrânse se plasează pe poziţiile cele mai de sus în listă, se obţine schema de codare din Fig. 3.6 şi graful şi cuvintele de cod ca în Fig. 3.7.

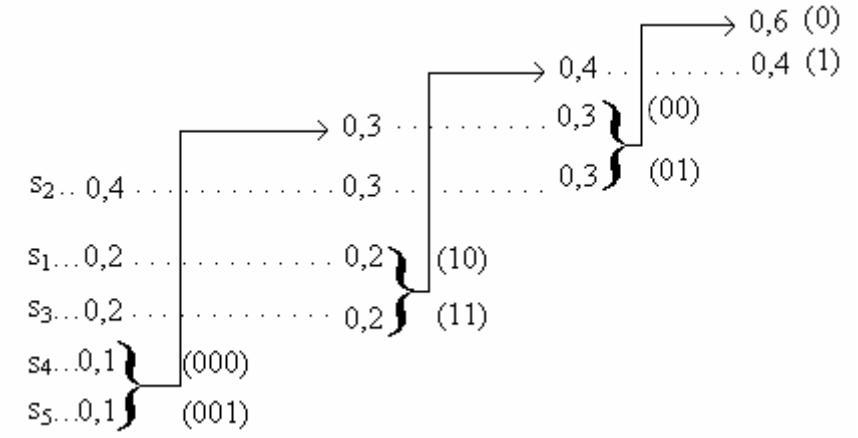


Fig. 3.6. Schema de codare

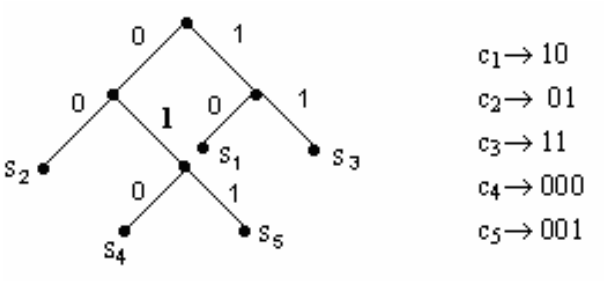


Fig. 3.7. Graful şi cuvintele de cod

Pentru acest cod, lungimea medie este, evident, aceeaşi, în timp ce dispersia devine

$$\sigma_2{}^2 = \sum_{i=1}^{5} \left(l_i - \overline{l} \right)^2 = 0,2^2 + 0,2^2 + 0,2^2 + 0,8^2 + 0,8^2 = 1,4$$

Deşi din punct de vedere informaţional, cele două coduri sunt identice, în practică se preferă folosirea celor de dispersie minimă, din motive de transmisie.

De exemplu, dacă se doreşte să se transmită mesaje ale sursei cu o viteză de 10.000 mesaje/sec., este necesar un canal cu capacitatea de 22.000 biţi/sec. Deoarece viteza de generare a biţilor oscilează în jurul valorii de 22.000 biţi/sec., funcţie de succesiunea de mesaje furnizate la un moment dat, ieşirea sursei este încărcată într-un buffer. Dacă, de exemplu, sursa generează la un moment dat şiruri de mesaje s_4 şi s_5 mai multe secunde, pentru primul cod se generează 40.000 biţi/sec. şi în fiecare secundă ar trebui un buffer de capacitate de 18.000 biţi. Cu al doilea cod se generează 30.000 biţi /sec. şi bufferul ar trebui să aibă capacitatea de 8.000 biţi. Dacă se transmit şiruri de mesaje s_2, cu primul cod se generează 10.000 biţi/sec. şi canalul nu e folosit la capacitatea sa, rămânând un deficit de 12.000 biţi/sec., pe când cu al doilea cod se generează 20.000

biţi/sec., deficitul existent în exploatarea canalului fiind numai de 2.000 biţi/sec. Aşadar, din motive de transmisie este mai rezonabil a se alege al doilea cod decât primul.

3.6.2. Procedeul de codare binară Shannon – Fano

Acest procedeu se aplică de obicei în cazurile particulare în care probabilităţile de furnizare ale mesajelor sunt puteri întregi pozitive ale lui (1/2), adică, de forma:

$$p(s_k)=\left(\frac{1}{2}\right)^{l_k}=2^{-l_k},\qquad (\forall)k=\overline{1,N} \tag{3.48}$$

unde l_k este un număr întreg pozitiv.

Dacă relaţia (3.48) este satisfăcută, mulţimea $S=\{s_1,s_2,...,s_N\}$ a mesajelor sursei discrete de informaţie ce urmează a fi codată poate fi partiţionată în două submulţimi S_0 şi S_1, astfel încât suma probabilităţilor mesajelor incluse în S_0, notată cu $p(S_0)$, să fie egală cu suma probabilităţilor mesajelor incluse în S_1, notată cu $p(S_1)$. Sursa S fiind totdeauna completă, se poate scrie:

$$\left.\begin{array}{l} p(S_0)=p(S_1) \\ p(S_0)+p(S_1)=1 \end{array}\right\} \rightarrow p(S_0)=p(S_1)=\frac{1}{2}=2^{-1} \tag{3.49}$$

Submulţimile S_0 şi S_1 se pot partiţiona la rândul lor în S_{00} şi S_{01}, respectiv în S_{10} şi S_{11}, astfel încât suma probabilităţilor mesajelor incluse în cele patru submulţimi să fie aceeaşi, adică se poate scrie relaţia:

$$p(S_{00})=p(S_{01})=p(S_{10})=p(S_{11})=\left(\frac{1}{2}\right)^2=2^{-2} \tag{3.50}$$

Se procedează în mod analog până se obțin submulțimi care conțin un singur mesaj. Se observă că fiecare submulțime are suma probabilităților mesajelor incluse egală cu o putere întreagă a lui (1/2). Puterea întreagă este egală cu numărul indicilor submulțimii respective Dacă submulțimea conține un singur mesaj, s_k, şi are un număr de indici egal cu l_k, atunci se poate scrie:

$$p(s_k) = \left(\frac{1}{2}\right)^{l_k} = 2^{-l_k} \tag{3.51}$$

de unde rezultă necesitatea ca sursa S ce urmează a fi codată să-şi furnizeze mesajele cu probabilități egale cu 1/2 la o putere întreagă, pozitivă.

Sursa fiind completă, se poate scrie:

$$\sum_{k=1}^{N} p(s_k) = 1 \tag{3.52}$$

Înlocuind (3.51) în (3.52), rezultă:

$$\sum_{k=1}^{N} 2^{-l_k} = 1, \tag{3.53}$$

ceea ce înseamnă că inegalitatea lui Kraft (3.1) devine în acest caz egalitate.

Cuvintele de cod se vor obține, atunci, astfel:

1. Se atribuie simbolul "0" submulțimii S_0 şi simbolul "1" submulțimii S_1, (sau invers), astfel că toate cuvintele corespunzătoare mesajelor incluse în S_0 vor începe cu "0" şi toate cuvintele corespunzătoare mesajelor incluse în S_1, vor începe cu "1" (sau invers);

2. Se alocă submulțimilor S_{00} şi S_{10} ca al doilea mesaj "0", iar submulțimilor S_{01} şi S_{11} ca al doilea mesaj "1" (sau invers). În felul acesta, cuvintele de cod corespunzătoare mesajelor incluse în

S_{00} vor începe cu 00, cuvintele de cod corespunzătoare mesajelor incluse în S_{10} vor începe cu 10 şi aşa mai departe, cuvintele de cod corespunzătoare mesajelor induse în S_{11} vor începe cu 11.

3. Operaţia se continuă în acelaşi mod, până când în fiecare submulţime rămâne un singur mesaj, căruia îi va corespunde cuvântul de cod format din şirul de indici ai submulţimii respective.

Deoarece la fiecare partiţionare în două submulţimi atribuirea mesajelor "0" şi "1" este arbitrară, rezultă că prin acest procedeu se pot obţine o multitudine de coduri instantanee, dar toate absolut optimale.

În principiu, procedeul de codare descris s-ar putea aplica în general, adică şi atunci când relaţia (3.53) nu este satisfăcută. În acest caz, partiţionările în submulţimi trebuie efectuate astfel încât suma probabilităţilor mesajelor incluse în submulţimile respective să fie cât mai apropiate. Atribuind simbolurile "0" şi "1" ca în procedeul descris, se obţin totdeauna coduri instantanee.

Cu cât sumele probabilităţilor mesajelor componente ale submulţimilor respective vor fi mai apropiate, cu atât lungimea medie a cuvintelor de cod va fi mai mică.

Exemplul 3.2.

Se consideră sursa discretă de informaţie caracterizată de distribuţia:

$$S:\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 & s_7 & s_8 \\ 2^{-2} & 2^{-2} & 2^{-3} & 2^{-3} & 2^{-4} & 2^{-4} & 2^{-4} & 2^{-4} \end{pmatrix}.$$

Procedeul de codare binară Shannon - Fano este sintetizat în tabelul de mai jos.

Mesaje	Probabilități	Partiții				Cuvânt de cod
s_1	2^{-2}	0	0			0 0
s_2	2^{-2}		1			0 1
s_3	2^{-3}	1	0	0		1 0 0
s_4	2^{-3}			1		1 0 1
s_5	2^{-4}		1	0	0	1 1 0 0
s_6	2^{-4}				1	1 1 0 1
s_7	2^{-4}			1	0	1 1 1 0
s_8	2^{-4}				1	1 1 1 1

Graful arborescent ataşat codului astfel obținut este reprezentat în Fig. 3.8.

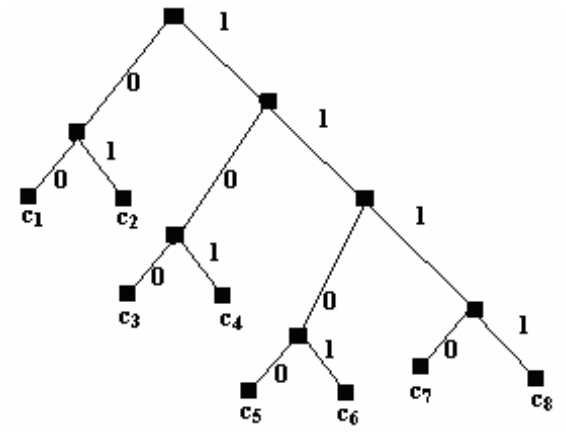


Fig. 3.8. Graful arborescent ataşat codului din tabel

3.6.3. Lungimea medie a cuvintelor de cod în cazul codului binar Huffman

Se ştie că pentru un cod optimal

$$H(S) \le \bar{l} \le H(S) + 1 \qquad (3.54)$$

Limita superioară din relația (3.54) nu este foarte precisă. În [25] se arată că dacă p_{max} este cea mai mare probabilitate cu care este furnizat un mesaj, atunci pentru p_{max}<0,5, limită superioară pentru codul Huffman este $H(S) + p_{\max}$, în timp ce pentru $p_{max} \ge 0{,}5$, limita superioară este $H(S) + p_{\max} + 0{,}086$.

În aplicațiile în care numărul de mesaje ale sursei este mare, p_{max} este relativ mic şi redundanța codului, care este diferența dintre lungimea medie şi entropie, exprimată în procente din entropie, este relativ mică. În cazul în care alfabetul sursei este mic şi probabilitățile de apariție ale diferitelor mesaje sunt neechilibrate, valoarea probabilității p_{max} poate fi relativ mare şi codul Huffman poate deveni ineficient, adică redundanța sa să reprezinte un procent semnificativ din entropie.

Exemplul 3.3.

Fie o sursă care furnizează mesajele $S = \{s_1, s_2, s_3\}$, cu probabilitățile $p(s_1) = 0{,}8;\ p(s_2) = 0{,}02;\ p(s_3) = 0{,}18$. Entropia sursei este de 0,816 biți/mesaj.

Codul Huffman pentru această sursă este dat în Fig. 3.9.

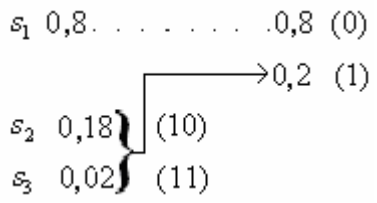


Fig. 3.9. Schema de codare pentru codul din Exemplul 3.3.

Lungimea medie pentru acest cod este 1,2 biți/simbol. Redundanța este 0,384 biți/simbol, aproximativ 47% din $H(S)$. Aceasta înseamnă că pentru a coda această sursă sunt necesari cu 47% mai mulți biți decât minimul necesar. Uneori, pentru micşorarea redundanței se realizează codarea Huffman pe extensia sursei inițiale. Astfel, dacă se consideră extensia de ordinul 2 a sursei precedente, se obțin 3^2=9 mesaje. Dacă se efectuează codarea Huffman a sursei extinse, se obțin cuvintele de cod din Tabelul 3.1.

Tabelul 3.1.

Mesaj compus	Probabilitate	Cuvânt de cod
s_1s_1	0,64	0
s_1s_2	0,016	10101
s_1s_3	0,144	11
s_2s_1	0,016	101000
s_2s_2	0,0004	101000
s_2s_3	0,0036	1010011
s_3s_1	0,1440	100
s_3s_2	0,0036	10100100
s_3s_3	0,0324	1011

Pentru acest cod, lungimea medie este $\bar{l}_e$=1,7516 biți/mesaj.

Cum un mesaj al sursei extinse este format din două mesaje ale alfabetului original, rezultă că lungimea medie, $\bar{l}$, exprimată în funcţie de alfabetul original este $\bar{l}$=0,8758, redundanţa este de aproximativ 0,06 biţi/simbol, adică aproximativ 7% din entropie.

Dacă probabilităţile mesajelor sunt mult neechilibrate, atunci ar putea fi necesară codarea pe extensii de ordin superior, în scopul scăderii la valori acceptabile ale redundanţei. Prin creşterea ordinului extensiei, mărimea alfabetului sursei extinse creşte exponenţial cu ordinul extensiei şi codarea Huffman poate deveni ineficientă din punct de vedere al preţului de cost, caz în care se foloseşte *codarea aritmetică*, care va fi discutată ulterior.

3.6.4. Procedeul de codare Huffman generalizat

În acest caz, alfabetul codului conţine mai mult de două simboluri. Procedeul de codare este asemănător celui din cazul binar, parcurgându-se următoarele etape:

1) Se ordonează mesajele sursei ce urmează a fi codată în ordinea descrescătoare a probabilităţilor;

2) Dacă alfabetul codului conţine $M \geq 3$ simboluri, se reunesc ultimele M mesaje (de probabilităţile cele mai mici) într-un singur mesaj, căruia i se alocă probabilitatea egală cu suma probabilităţilor mesajelor componente. Se ordonează din nou mesajele în ordinea descrescătoare a probabilităţilor, formându-se astfel prima sursă restrânsă R_1. Procedându-se în mod analog, se formează sursa restrânsă R_2 din R_1, R_3 din R_2 şi aşa mai departe, până când se obţine o sursă restrânsă care conţine M mesaje. Pentru ca ultima sursă restrânsă să conţină M mesaje cărora să li se aloce arbitrar cele M mesaje din alfabetul codului, înainte de a realiza

restrângerile respective, se face următorul raționament:

- la formarea primei surse restrânse, reunindu-se M mesaje într-un singur mesaj, va rezulta un număr de mesaje egal cu $N-M+1=N-(M-1)$;

- a doua sursă restrânsă va conține, prin reunirea ultimelor M mesaje, un număr de mesaje egal cu $N-2M+2=N-2(M-1)$;

- raționându-se în mod analog, după n restrângeri, ultima sursă restrânsă va conține un număr de mesaje egal cu $N-n(M-1)$, care trebuie să fie egal cu numărul M al mesajelor din alfabetul codului, adică

$$M=N-n(M-1)\Rightarrow n=\frac{N-M}{(M-1)} \tag{3.55}$$

Deoarece n (numărul de restrângeri) trebuie să fie un număr întreg pozitiv, se va considera:

$$n_1=\left\lceil \frac{N-M}{M-1} \right\rceil \tag{3.56}$$

unde $\lceil m \rceil$ simbolizează cel mai mic număr întreg, mai mare sau egal decât m. Astfel, sursa va trebui să aibă un număr de mesaje N_1, calculat cu relația

$$N_1=M+n_1(M-1) \tag{3.57}$$

- Dacă sursa S ce urmează a fi codată are un număr N de mesaje care nu verifică relația (3.57), se va adăuga la sursa respectivă un număr de mesaje, până când această relație este satisfăcută. Mesajelor adăugate li se vor aloca probabilități nule, astfel că sursa inițială nu va fi alterată, deoarece mesajele de probabilități nule nu vor fi furnizate niciodată.

3) La fiecare partiție în M submulțimi se alocă arbitrar cele M mesaje din alfabetul codului. Deoarece alocarea celor M mesaje din

alfabetul codului se face arbitrar, rezultă că prin acest procedeu va rezulta o multitudine de coduri instantanee, toate cu aceeaşi lungime medie a cuvintelor de cod, care nu mai poate fi micşorată prin nici un alt procedeu de codare, adică toate codurile astfel obținute vor fi instantanee şi compacte.

Exemplul 3.4.

Se presupune sursa discretă de informație caracterizată de distribuția

$$S:\begin{pmatrix} s_1 & s_2 & s_3 & s_4 & s_5 & s_6 \\ 0,1 & 0,2 & 0,3 & 0,15 & 0,05 & 0,2 \end{pmatrix}$$

Dacă alfabetul codului este $X=\{x_1,x_2,x_3\}$, să se realizeze o codare Huffman generalizată.

Înainte de a realiza codarea, se verifică dacă este satisfăcută relația (3.57) care, pentru cazul particular considerat, devine: $N_1=2n_1+3$, unde $n_1=\left\lceil\frac{6-3}{3}\right\rceil=2$, deci $N_1=7$. Va trebui adăugat un nou mesaj, fie acesta s_7, de probabilitate nulă, adică $p(s_7)=0$. Pentru realizarea codării, se procedează după cum se arată în Fig. 3.10.

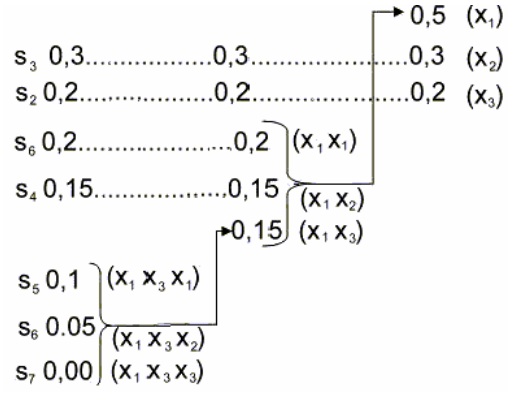


Fig. 3.10. Schema de codare pentru Exemplul 3.4.

Cuvintele de cod corespunzătoare mesajelor sursei sunt

$$s_1 \to c_1 \to x_1x_3x_1$$
$$s_2 \to c_2 \to x_3$$
$$s_3 \to c_3 \to x_2$$
$$s_4 \to c_4 \to x_1x_2$$
$$s_5 \to c_5 \to x_1x_3x_2$$
$$s_6 \to c_6 \to x_1x_1$$

Graful corespunzător codului este dat în Fig. 3.11.

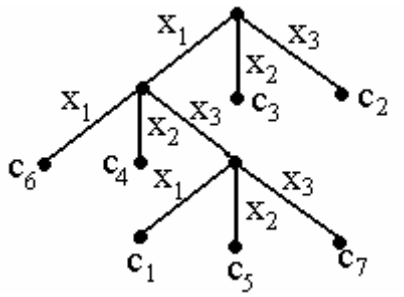


Fig. 3.11. Graful ataşat codului

3.7. Codarea binară Huffman dinamică (adaptivă)

Procedeul de codare binară Huffman descris în paragraful 3.6 necesită cunoaşterea probabilităţilor cu care sursa îşi furnizează mesajele. În literatura de specialitate această situaţie este cunoscută şi sub denumirea de codare Huffman statică. În cazul în care probabilităţile de furnizare a mesajelor nu sunt cunoscute, se foloseşte codarea Huffman *dinamică* sau *adaptivă*.

În descrierea codării Huffman dinamice sau adaptive se vor

folosi următoarele notaţii şi noţiuni:

- Nodurile terminale sau externe din graful arborescent se vor numi frunze, corespunzând mesajelor sursei;

- Cuvântul de cod pentru un mesaj se obţine parcurgând arborele de la rădăcină la frunza corespunzătoare simbolului. Prin convenţie, zero se va aloca unei ramuri din stânga şi unu, unei ramuri din dreapta;

- Nodurile de la extremităţile ramurilor care pleacă dintr-un nod reprezintă fiii sau copiii nodului respectiv, numit nod părinte;

- Ponderea unui nod extern este numărul de apariţii a simbolului corespunzător frunzei respective până la acel moment;

- Ponderea unui nod intern este suma ponderilor fiilor nodului respectiv;

- Dacă sursa ce urmează a fi codată furnizează n mesaje, în graf există $2n+1$ noduri interne şi externe, numerotate în continuare $y_1,...,y_{2n+1}$. Dacă x_j este ponderea nodului cu numărul y_j, trebuie să existe relaţia $x_1 \le x_2 \le ... \le x_{2n+1}$;

- Nodurile y_{2j-1} şi y_{2j} sunt fii ai aceluiaşi nod părinte, iar numărul de ordine al părintelui este mai mare decât y_{2j-1} şi y_{2j}.

Ultimele două caracteristici se numesc de *fraternitate* şi orice arbore care are această proprietate este un *arbore Huffman.*

În codarea Huffman dinamică, nici emiţătorul, nici receptorul nu cunosc statistica sursei la începerea transmisiei, astfel încât arborii de la transmisie şi recepţie constau dintr-un singur nod corespunzător mesajelor încă netransmise, de pondere zero. Pe parcursul transmisiei, nodurile corespunzătoare mesajelor transmise sunt adăugate arborelui, care este reconfigurat pe baza unui procedeu de actualizare. Înaintea începerii transmiterii se stabileşte un cod pentru fiecare mesaj, după cum urmează:

Dacă sursa furnizează mesajele $s_1, s_2, ..., s_N$, se determină parametrii e şi r, astfel încât $N = 2^e + r, 0 \le r < 2^e$. Mesajul s_k este codat prin reprezentarea pe $e+1$ biți a lui $k-1$, dacă $1 \le k \le 2r$, în caz contrar, s_k este reprezentarea pe e biți a lui $k-r-1$. De exemplu, pentru $N = 26 \to r = 10$ şi $e = 4 \to s_1 = 00000$, $s_2 = 00001, ..., s_{22} = 1011$.

3.7.1. Actualizarea arborelui

Algoritmul de actualizare necesită ca nodurile să fie numerotate într-o ordine stabilită. Cel mai mare număr de nod este al rădăcinii. Cel mai mic se asignează nodului încă netransmis, care este nod gol, (NG).

Mulțimea nodurilor cu aceeaşi pondere formează un *bloc*. Rolul algoritmului de actualizare este de a păstra proprietatea de *fraternitate* în timpul modificării arborelui, pentru a reflecta ultima estimare a frecvenței de furnizare a mesajului.

Pentru ca procedura de actualizare de la emițător şi receptor să opereze cu aceeaşi informație, arborele de la emițător este actualizat după codarea fiecărui mesaj, iar la recepție arborele este actualizat după decodarea fiecărui mesaj.

Organigrama algoritmului de actualizare este prezentată în Fig. 3.12.

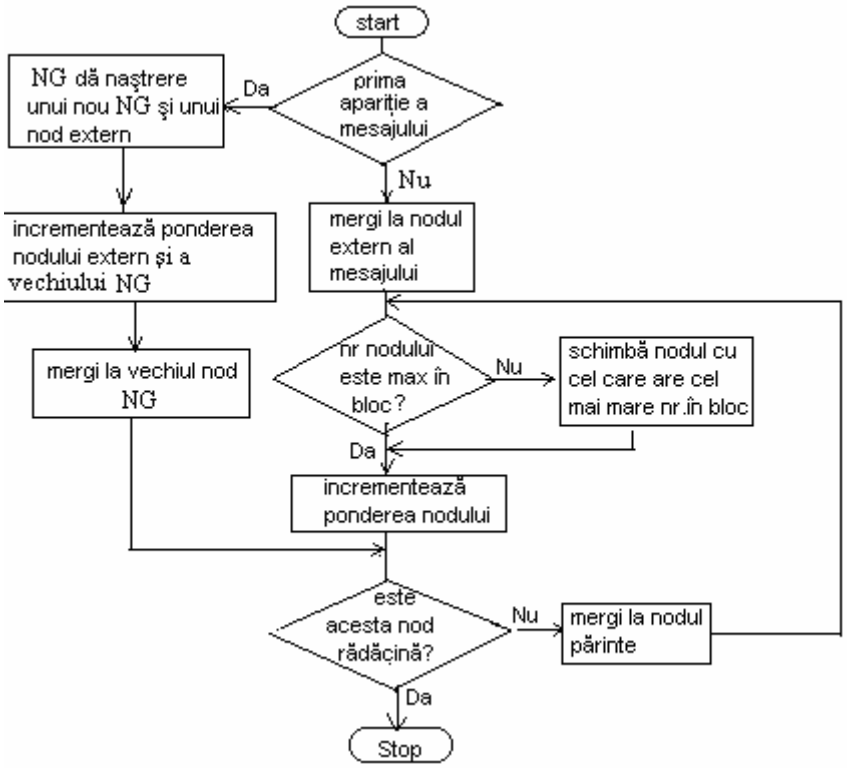


Fig. 3.12. Procedura de actualizare pentru algoritmul de codare dinamică Huffman

Procedura de actualizare este următoarea: după ce un mesaj a fost codat la emisie sau decodat la recepție, nodul extern corespunzător mesajului este examinat pentru a vedea dacă are cel mai mare număr de nod din bloc. Dacă nodul extern nu are cel mai mare număr de nod, el este interschimbat cu nodul care are cel mai mare număr din bloc, cu condiția ca acesta să nu fie părintele nodului ce urmează a fi actualizat. Ponderea nodului extern este apoi incrementată. Dacă nu se interschimbă nodurile înaintea incrementării ponderii nodului, este posibil ca ordonarea cerută de proprietățile de fraternitate să fie distrusă. Odată incrementată ponderea nodului, s-a adaptat arborele Huffman la acel nivel.

Se trece apoi la nivelul următor, prin examinarea nodului părinte al nodului a cărui pondere a fost incrementată, pentru a vedea dacă are cel mai mare număr din bloc. Dacă nu are, este schimbat cu nodul cu cel mai mare număr din bloc. Din nou, face excepție nodul cu cel mai mare număr, dacă acesta este părinte pentru nodul considerat. Odată ce s-a efectuat schimbarea (sau s-a decis că nu este necesară), ponderea nodului părinte este incrementată. Procesul se termină când se ajunge la nodul rădăcină.

Când mesajul ce urmează a fi codat sau decodat apare pentru prima dată, acestuia i se atribuie un nod extern şi se ataşează un nou nod NG în graf.

Atât nodul extern cât şi nodul NG sunt fii al vechiului NG. Cum mesajul corespunzător noului nod extern a apărut o singură dată, acestuia i se atribuie ponderea 1.

Cum vechiul nod NG este părinte pentru noul nod extern, ponderea sa se incrementează cu 1 şi se continuă astfel până la rădăcină.

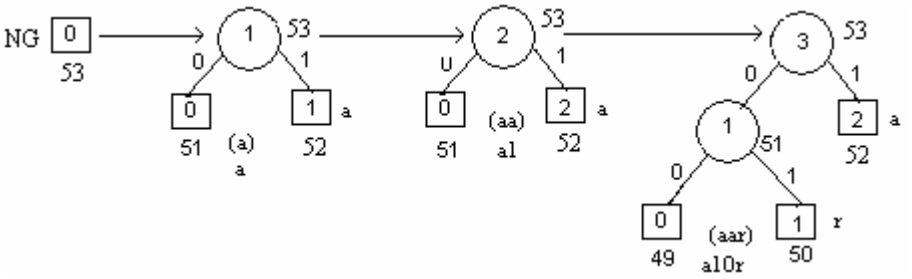
Exemplul 3.5.

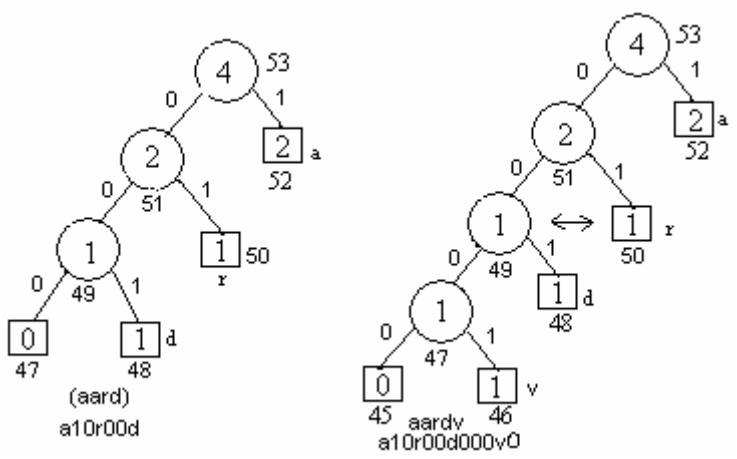
Se presupune că se doreşte codarea Huffman adaptivă a secvenței $[a\ a\ r\ d\ v\ a\ ]$, care conține 4 din 26 de mesaje posibile, reprezentate de literele mici ale unui alfabet. Pentru codarea acestor mesaje se foloseşte codul prezentat în paragraful 3.7.

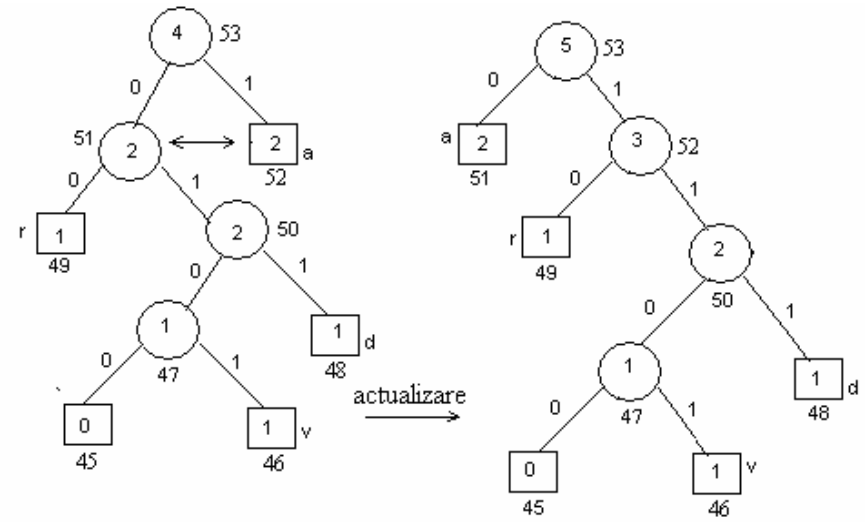
Procesul de actualizare este arătat în Fig. 3.13. Se începe cu nodul NG. Numărul total de noduri este $2 \cdot 26 + 1 = 53$, astfel încât se începe numărătoarea inversă, cu numărul 53 asignat nodului rădăcină. Prima literă ce urmează a se transmite este *a*. Cum litera nu este în arbore, se transmite *a* şi se adaugă aceasta în arbore. Nodul NG dă naştere unui nou nod NG şi unui nod terminal, corespunzător literei *a*. Ponderea nodului terminal va fi mai mare

decât a nodului NG, astfel încât se asignează numărul 51 nodului NG şi 52 nodului terminal corespunzător literei *a*. Următoarea literă de transmis este tot *a*. De această dată, codul transmis este 1, deoarece litera este în graf. Nodul corespunzător lui *a* are cel mai mare număr din bloc (dacă nu se consideră nodul său părinte), aşa încât nu este nevoie să schimbăm nodurile. Se incrementează ponderea nodului extern şi a nodului rădăcină. Următoarea literă de transmis este *r*. Cum litera nu are un nod corespunzător în arbore, se transmite cuvântul de cod pentru nodul NG, care este 0, urmat de indexul lui *r*. Nodul NG dă naştere unui nou nod NG şi unui nod extern corespunzător lui *r*. Din nou, nu este necesară actualizarea arborelui, ci numai incrementarea corespunzătoare a ponderilor nodurilor. Următoarea literă este *d*, care, de asemenea, se transmite pentru prima dată. Se transmite codul pentru nodul NG, care este 00, urmat de indexul pentru *d*. Nodul NG dă iar naştere la două noduri. Încă nu este nevoie de a actualiza arborele. Acesta se schimbă cu transmiterea literei următoare, *v*, care nu a mai fost întâlnită. S-a transmis codul frunzei goale, 000, urmat de codul pentru litera *v*. În arbore se adaugă nodurile 45 şi 46, cu 46 ca nod terminal pentru *v*. Se incrementează ponderea nodului extern 46 şi a vechiului NG. Se merge la nodul părinte al vechiului NG, adică nodul 49 care nu are cel mai mare număr din bloc, aşa că se schimbă cu 50, care este cel mai mare număr din bloc şi se incrementează ponderea acestuia. Se merge la rădăcina nodului 50 care este 51 şi care nu are cel mai mare număr din bloc, aşa că se schimbă cu 52, a cărui pondere se incrementează. Se merge la rădăcina nodului 52, adică nodul 53, care este nod rădăcină şi se incrementează ponderea acestuia. Urmează mesajul *a*, care este în graf, aşa încât se transmite 0. Deoarece numărul nodului extern corespunzător lui *a* este maxim în bloc, se incrementează ponderea acestuia şi a părintelui, care este

nod rădăcină.







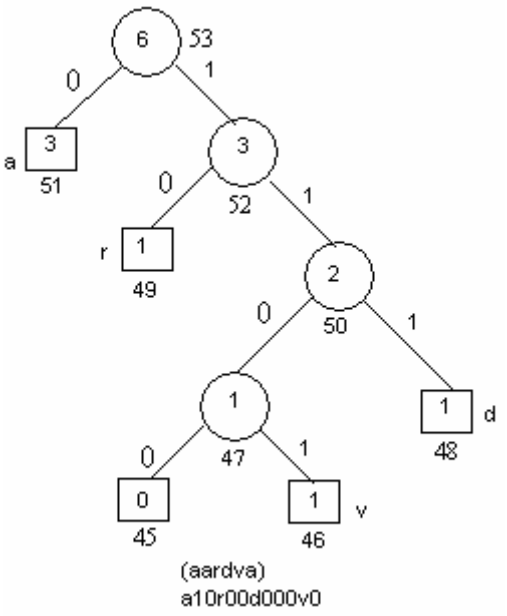


Fig. 3.13. Arborele Huffman adaptiv, după ce s-a transmis secvența *aardva*
a10r00d000v0

3.7.2. Codarea

Diagrama pentru codare este prezentată în Fig. 3.14. Inițial, arborii de la codare şi decodare sunt formați dintr-un singur nod NG. Prin urmare, cuvântul de cod corespunzător primului mesaj ce apare este transmis în clar. După primul mesaj, de fiecare dată când trebuie codat un mesaj ce apare pentru prima dată, se transmite întâi codul pentru nodul NG, care se obține prin parcurgerea arborelui Huffman de la rădăcină la nodul NG.

Aceasta înştiințează receptorul că mesajul al cărui cod urmează nu are încă un nod în arborele Huffman. Codul nodului NG este urmat de codul prestabilit pentru mesaj. Dacă mesajul ce urmează a fi codat are un nod corespunzător în arbore, atunci este transmisă succesiunea de biți obținută prin traversarea arborelui de

la rădăcină la nodul extern corespunzător mesajului.

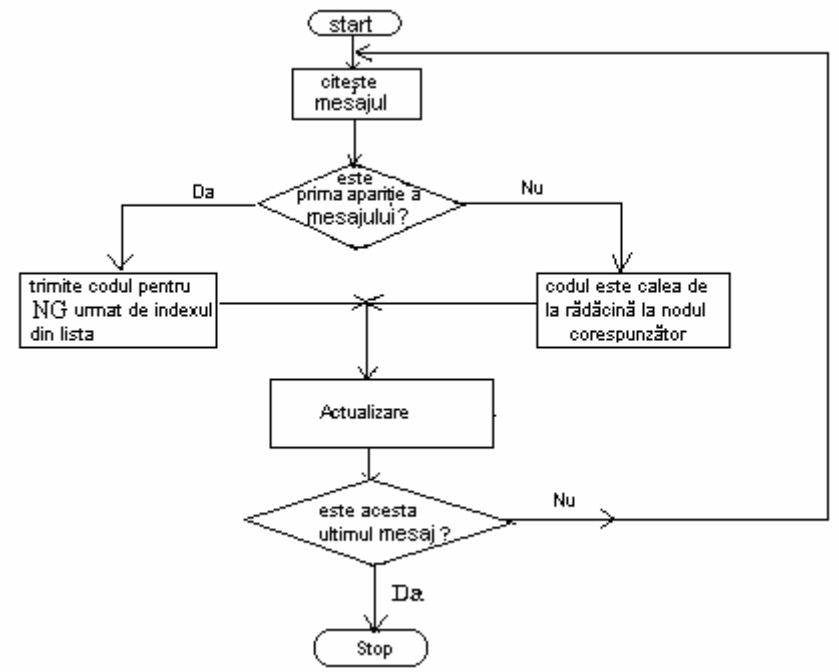


Figura 3.14. Organigrama pentru codare

Exemplul 3.6.

În Exemplul 3.5 s-a folosit un alfabet de 26 de litere. Pentru a obține codul prestabilit pentru mesaj trebuie găsite valorile pentru *e* şi *r*, astfel încât $2^e + r = 26$ cu $0 \le r < 2^e \rightarrow e = 4, r = 10$.

Primul mesaj de codat este *a*. Cum *a* este prima literă din alfabet, k=1. Deoarece $2 \cdot r = 20 > k = 1$, litera *a* va fi reprezentarea pe $e + 1 = 5$ biți a numărului $k - 1 = 1 - 1 = 0$, adică 00000. Arborele este actualizat după diagrama din Fig. 3.13, nodul gol a dat naştere unui nod extern corespunzător lui *a* şi unui nou nod gol. Nodul extern corespunzător lui *a* are ponderea 1, iar nodul gol, 0. Nodul intern are ponderea 1 (suma ponderilor fiilor). Următorul mesaj este

a. Cum acesta există deja în arbore, cuvântul de cod corespunzător se obține prin traversarea grafului de la rădăcină la nodul corespunzător, în cazul de față este 1.

După transmiterea codului pentru cel de-al doilea *a*, ponderea nodului extern corespunzător acestuia este incrementată la 2, ca şi ponderea părintelui.

Al treilea mesaj de transmis este *r*. Cum acesta este la prima apariție, se transmite codul nodului gol, adică 0, urmat de codul lui *r*. (*r,* fiind a optsprezecea literă din alfabetul de 26 de litere, înseamnă că $k = 18$; deoarece $18<2r = 20$, litera *r* va fi reprezentarea binară pe $k+1=4+1=5$ biți a numărului $k-1=18-1=17$, adică 10001). Arborele este actualizat şi se continuă cu mesajul *d*. Folosind acelaşi procedeu pentru *d,* se transmite codul nodului gol, care este 00 urmat de indexul lui *d*, care este 00011 (*d* fiind a patra literă din alfabet şi deoarece $4 < 20$, litera *d* va fi reprezentarea binară pe $k+1=4+1=5$ biți a numărului $k-1=4-1=3$, adică 00011). Următorul simbol, *v* , este al 22-lea în alfabet şi fiind mai mare decât 20 se transmite codul pentru nodul gol, 000, urmat de reprezentarea binară pe 4 biți a lui 22-10-1=11, adică 1011. Următorul mesaj este *a*, pentru care se transmite 0.

Secvența transmisă este:

a 1 0 *r* 0 0 *d* 0 0 0 *v* 0 , echivalentă cu

00000	1	0	1001	00	00011	000	1011	0
a	a	gol	r	gol	d	gol	v	a

3.7.3. Decodarea

Organigrama pentru decodare este prezentată în Fig. 3.15.

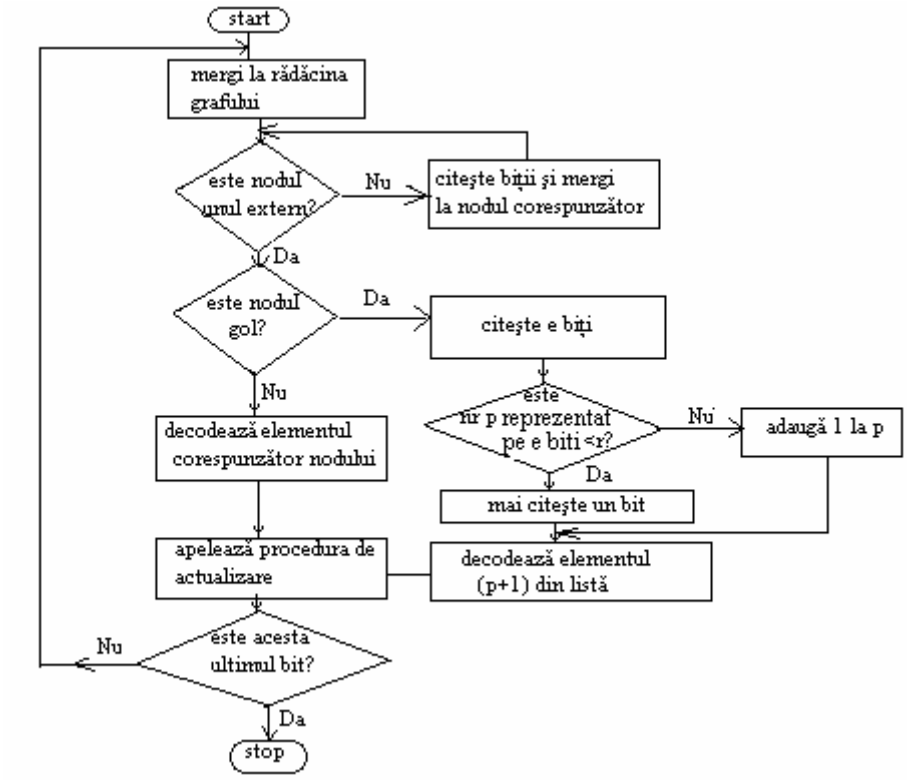


Fig. 3.15. Organigrama pentru decodarea Huffman dinamică

Pe măsură ce este citită secvenţa binară recepţionată, arborele se parcurge într-un mod identic cu cel de la codare.

Odată identificat codul unei frunze din secvenţa recepţionată, se decodează mesajul corespunzător acelei frunze. Dacă frunza este frunză goală, se verifică următorii *e* biţi pentru a vedea dacă rezultă un număr mai mic decât *r*. Dacă este mai mic, se mai citeşte un bit pentru a completa codul mesajului. Indexul mesajului se obţine adăugând o unitate la numărul zecimal corespunzător secvenţei de *e* sau *e*+1 biţi. Odată decodat mesajul, arborele este actualizat şi următorul bit recepţionat este folosit pentru a porni o nouă parcurgere a arborelui.

Exemplul 3.7.

Să se decodeze secvența codată anterior, 0 0 0 0 0 1 0 1 0 0 0 1 0 0 0 0 0 1 1 0 0 0 1 0 1 1 0.

Inițial arborele de la decodor constă numai dintr-un nod gol, aşa că primul mesaj ce urmează a fi decodat se obține din listă. Se citesc *e*=4 biți, 0000, care corespunde numărului zecimal 0. Deoarece 0<r =10 rezultă că se mai citeşte un bit, rezultând secvența 00000. Se decodează în zecimal şi se adaugă o unitate, astfel indexul simbolului recepționat este 1, care îi corespunde lui *a* în listă, prin urmare, prima literă decodată este *a.*

Arborele este actualizat. Următorul bit recepționat este 1, care îi corespunde căii de la rădăcină la nodul extern *a,* astfel încât al doilea mesaj decodat este tot *a*.

Următorul bit este 0, care corespunde căii de la rădăcină la nodul gol. Următorii patru biți 1000 corespund numărului zecimal 8, care este mai mic decât 10, astfel încât se mai citeşte un bit pentru a obține cuvântul 10001, al cărui echivalent zecimal plus o unitate este 18, care este indexul lui *r*. Se decodează *r* şi se actualizează arborele. Următorii doi biți 00 reprezintă parcurgerea grafului până la frunza goală. Se citesc următorii 4 biți 0001, care corespund lui 1<10, deci se mai citeşte un bit, adică secvența 00011. Pentru a obține indexul simbolului recepționat din listă, se adaugă o unitate la valoarea zecimală a celor 5 biți. Valoarea indexului este 4, care corespunde simbolului *d*. Continuând, se obține secvența decodată *a a r d v a.*

3.8. Aplicații ale codării Huffman

În continuare sunt prezentate câteva aplicații simple ale codării Huffman în compresie, această codare fiind, de obicei, folosită în practică împreună cu alte tehnici de codare.

1. Compresia fără pierderi a imaginilor

Codarea Huffman poate fi folosită în compresia imaginilor digitale, când fiecare pixel poate lua valori dintr-o mulțime finită. În cazul imaginilor monocrome, această mulțime constă din numere întregi de la 0 la 225.

Rezultatele pentru patru imagini de test pentru care s-a aplicat codarea Huffman sunt arătate în Tabelul 3.2 [86].

Tabelul 3.2

Numele imaginii de test	Biți/pixel	Mărimea totală a imaginii(biți)	Raportul de compresie
Sena	6,90	56.525	1,16
Sensin	7,38	60.457	1,08
Pământ	4,78	39.158	1,67
Omaha	7,00	57.488	1,14

În reprezentarea imaginii originale (necompresate) se folosesc 8 biți/pixel. Imaginea constă din 256 de rânduri a câte 256 de pixeli, deci reprezentarea necompresată foloseşte 65.536 biți.

Din Tabelul 3.2 se observă că raportul de compresie diferă de la imagine la imagine, obținându-se o reducere de numai aproximativ ½ până la 1bit/pixel, după compresie. Pentru unele

aplicații această reducere este acceptabilă. De exemplu, dacă într-o arhivă există sute de imagini, o reducere de un bit pe pixel salvează mulți megabyți în spațiul discului. În codarea din exemplul considerat nu s-a ținut cont de structura din date, dată de corelația dintre pixeli. Dintr-o inspecție vizuală a imaginii, se poate observa că într-o imagine pixelii sunt corelați cu vecinii lor. Un model grosier pentru determinarea valorii unui pixel poate fi dat de relația $\hat{x}_n = x_{n-1}$, adică valoarea estimată pentru un pixel este egală cu a valoarea pixelului precedent. Secvența reziduală va fi diferența dintre pixelii vecini. Dacă se consideră acest model şi se foloseşte codul Huffman pentru a coda secvența reziduală, rezultatele sunt cele din Tabelul 3.3 [86].

Tabelul 3.3.

Numele imaginii de test	Biți/pixel	Mărimea totală a imaginii (biți)	Raportul de compresie
Sena	3,84	31.472	2,08
Sensin	4,63	37.880	1,73
Pământ	3,92	32.136	2,04
Omaha	6,34	51.986	1,26

După cum se observă, se obține o îmbunătățire a raportului de compresie față de cazul precedent. Rezultatele din Tabelele 3.2 şi 3.3 sunt obținute folosind un sistem în doi paşi, în unul s-a obținut statistica sursei, iar în al doilea s-a efectuat codarea Huffman. În loc să se folosească sistemul în doi paşi, se poate folosi un cod Huffman adaptiv. Rezultatele pentru acesta sunt prezentate în Tabelul 3.4 [86].

Tabelul 3.4.

Nume imagine de test	Biți/pixel	Mărimea totală a imaginii (biți)	Raportul de compresie
Sena	3,93	32,261	2,03
Sensin	4,63	37,896	1,73
Pământ	4,82	39,504	1,66
Omaha	6,39	52,321	1,25

Se observă că diferențele dintre codul Huffman static şi cel adaptiv sunt mici. Algoritmul Huffman adaptiv este preferat în sistemele care lucrează on-line sau în timp real, când nu este posibilă cunoaşterea prealabilă a statisticii sursei. Acesta, însă, este mai dificil de implementat decât cel static. Aplicația particulară va determina care din cele două este mai convenabil.

2. Compresia textelor

În compresia textelor se foloseşte frecvent codarea Huffman, deoarece în texte, se foloseşte un alfabet discret care, într-un domeniu dat, are probabilități relativ staționare. De exemplu, modelul de probabilitate pentru un roman nu va diferi semnificativ de modelul de probabilitate pentru alt roman. Similar, modelul de probabilitate pentru un set de programe C nu va diferi prea mult de modelul de probabilitate pentru un alt set de programe C.

O îmbunătățire a compresie se poate obține, dacă mai întâi se îndepărtează redundanța din date, materializată în corelația între mesajele din fişier, care este semnificativă în textele literare. De exemplu, în capitolul de față, *Huf* este întotdeauna urmat de *fman.*

3. Compresia audio

Altă clasă de date care se pretează foarte bine pentru compresie este clasa de date audio de pe compact discuri (CD). Semnalul audio pentru fiecare canal stereo este eşantionat la 44,1 kHz şi fiecare eşantion este reprezentat pe 16 biţi, ceea ce conduce la o cantitate enormă de date stocate pe CD care, pentru a fi transmise, ar necesita un canal cu capacitate semnificativă. Compresia este cu siguranţă utilă în acest caz. În Tabelul 3.5se arată, pentru diverse tipuri de surse audio, mărimea fişierului, entropia, mărimea estimată a fişierului compresat cu codarea Huffman şi raportul de compresie rezultat [86].

Tabelul 3.5. Codarea Huffman pe 16 biţi a semnalului audio

Numele fişierului	Mărimea originală a fişierului	Entropia (biţi)	Mărimea estimată a fişierului compresat(biţi)	Raportul de compresie
Mozart	939.862	12,8	725.420	1,30
Cohn	402.442	13,8	349.300	1,15
Mir	884.020	13,7	759.540	1,16

Ca şi la alte aplicaţii, se poate obţine o creştere a compresiei, dacă mai întâi este îndepărtată corelaţia din date. Datele audio pot fi modelate numeric. Se foloseşte modelul foarte simplu care a fost folosit în exemplul codării imaginii: valoarea estimată a unui eşantion este aceeaşi cu valoarea eşantionului anterior. Folosind acest model se obţine o secvenţă reziduală. Entropia acestei secvenţe diferenţă este prezentată în Tabelul 3.6 [86].

Tabelul 3.6. Codarea Huffman pe 16 biți a secvenței reziduale audio

Numele fişierului	Mărimea originală a fişierului (bytes)	Entropia diferențială (biți)	Mărimea estimată a fişierului compresat (bytes)	Raportul de compresie
Mozart	939.862	9,7	569.792	1,65
Cohn	402.442	10,4	261.590	1,54
Mir	884.020	10,9	602.240	1,47

Se observă o reducere de aproximativ 60% a mărimii fişierului compresat față de fişierul original.