



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

4. Coduri unice decodificabile

Coduri unic decodificabile

- Codificarea sursei S , prin alfabetul X , înseamnă găsirea unei corespondențe biunivoce $\gamma(S, X)$ între cuvintele de cod $\{c_i / i=1, 2, \dots, N\}$ și simbolurile sursei S
- Fiecare cuvânt de cod este o succesiune finită de litere din alfabetul codului
- Probabilitatea unui cuvânt de cod este dată de probabilitatea simbolului care trebuie transmis
- **Definiție:** Un cod se numește *unic decodabil* dacă fiecărei succesiuni de cuvinte de cod îi corespunde o singură succesiune de mesaje (simboluri) ale sursei.
- **Definiție:** Dacă un cuvânt de cod c_i este format din succesiunea $x_{i1}x_{i2}x_{im}$, atunci succesiunea $x_{i1} \dots x_{ik}$, $k \leq m$ se numește *prefix* al cuvântului respectiv
- Dacă nici un cuvânt de cod nu este prefix pentru alt cuvânt de cod se spune că acel cod are proprietatea de prefix.

Coduri unic decodificabile

- Exemple
 - Alfabet = $\{a_1, a_2, a_3, a_4\}$
 - $P(a_1) = 1/2, P(a_2) = 1/4, P(a_3) = P(a_4) = 1/8$
 - $H = 1.75$ bits
 - $n(a_i) = \text{lungime}(\text{cuvant cod}(a_i)), i=1..4$
 - Lungime medie $l = \sum_{i=1..4} P(a_i) n(a_i)$
- Coduri posibile:

	Probabilitate	Cod 1	Cod 2	Cod 3	Cod 4
a_1	0.500	0	0	0	0
a_2	0.250	0	1	10	01
a_3	0.125	1	00	110	011
a_4	0.125	10	11	111	0111
l		1.125	1.250	1.750	1.875

Coduri unic decodificabile

	Probabilitate	Cod 1	Cod 2	Cod 3	Cod 4
a_1	0.500	0	0	0	0
a_2	0.250	0	1	10	01
a_3	0.125	1	00	110	011
a_4	0.125	10	11	111	0111
l		1.125	1.250	1.750	1.875

- Cod 1
 - Coduri identice pentru a_1 si $a_2 \implies \text{decode}('00') = ???$
- Cod 2
 - Coduri unice dar ambigue: $\text{decode}('00'/'11') = ???$
- Cod 3
 - Unic decodificabil, instantaneu
- Code 4
 - Unic decodificabil, aproape instantaneu

Coduri unic decodificabile

- Decodificare unica:
 - Pentru orice secventa de cuvinte cod exista o unica decodificare pentru ea
- Unica \neq instantanee
 - Ex.:
 - $a_1 \Leftrightarrow 0$
 - $a_2 \Leftrightarrow 01$
 - $a_3 \Leftrightarrow 11$
 - $\text{decode}(011111111) = a_1a_3\dots$ or $a_2a_3\dots$?
 - Nu se stie pana la sfarsitul secventei
 - $011111111 \rightarrow 01111111a_3 \rightarrow 011111a_3a_3 \rightarrow 0111a_3a_3a_3 \rightarrow 01a_3a_3a_3a_3 \rightarrow a_2a_3a_3a_3a_3$

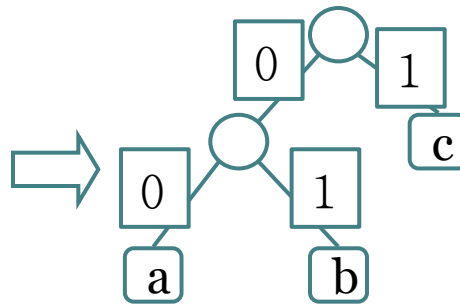
Test de Decodificare Unica

- Prefix si sufix
 - Fie $a = a_1 \dots a_k$, $b = b_1 \dots b_n$ coduri binare si $k < n$
 - Daca $a_1 \dots a_k = b_1 \dots b_k$ atunci a este **prefix** al lui b si
 - $b_{k+1} \dots b_n$ este **sufix** (dangling suffix) al lui a : $ds(a, b)$
- Algoritm
 - Fie $C = \{c_n\}$ setul tuturor cuvintelor cod
 - Pentru toate perechile (c_i, c_j) in C repeta:
 - Daca $ds(c_i, c_j) \notin C$ // sufixul NU este cuvânt cod
 - $C = C \cup ds(c_i, c_j)$
 - Altfel // sufixul este cuvânt cod
 - return NOT_UNIQUE
 - pana cand nu mai exista perechi unice
 - return UNIQUE

Coduri Prefix

- Cod prefix:
 - Niciun cuvânt cod nu este prefix al altuia
- Arborii binari = decodificatori de prefix:

symbol	code
a	00
b	01
c	1



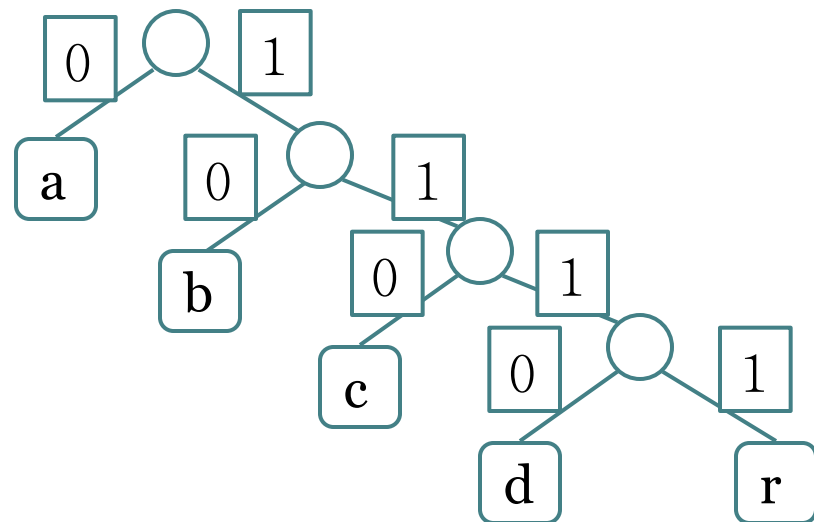
```

repeat
curr = root
  repeat
    if get_bit(input) = 1
      curr = curr.right
    else
      curr = curr.left
  until is_leaf(curr)
  output curr.symbol
until eof(input)

```

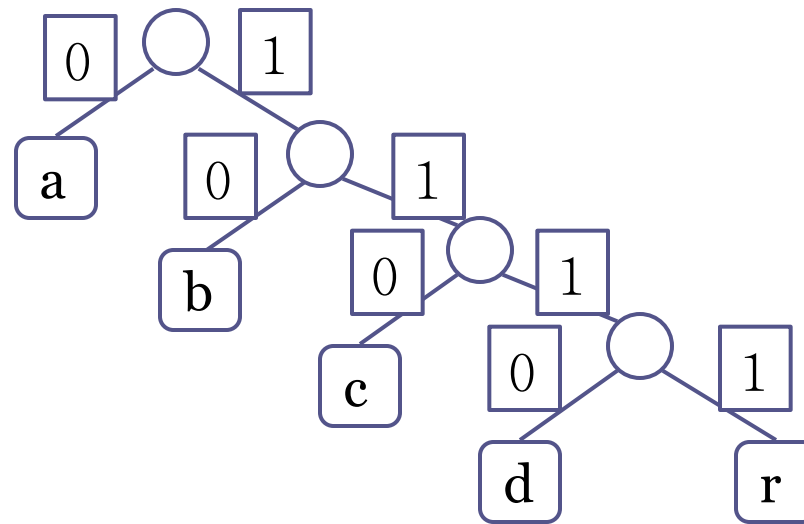
Decodificarea codurilor prefix. Exemplu

simbol	cod
a	0
b	10
c	110
d	1110
r	1111



abracadabra = 00111101100111001011110

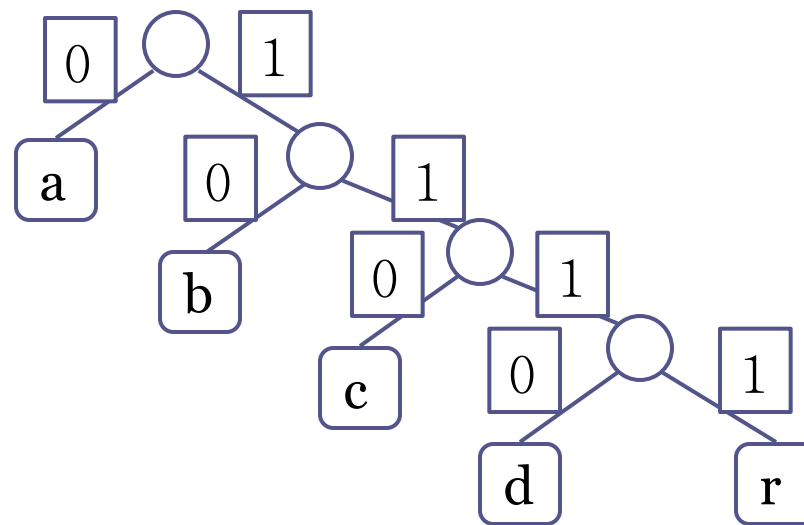
Exemplu decodificare



Input = 010111101100111001011110

Output = -----

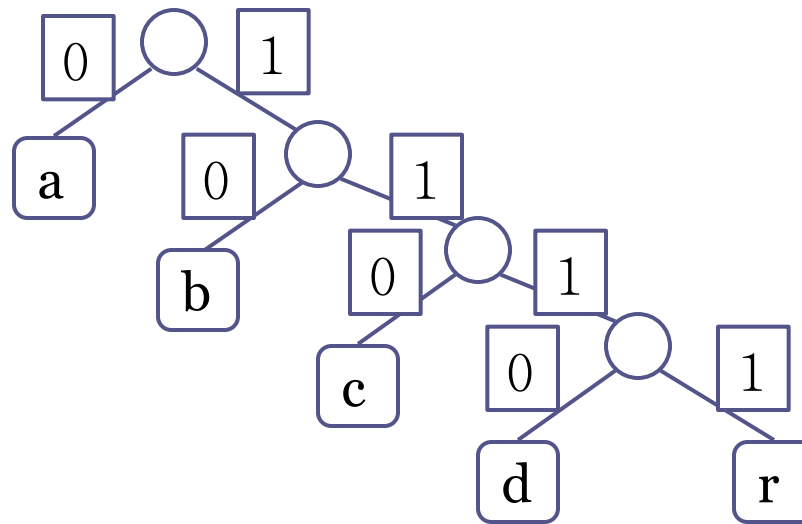
Exemplu decodificare



Input = 010111101100111001011110

Output = a-----

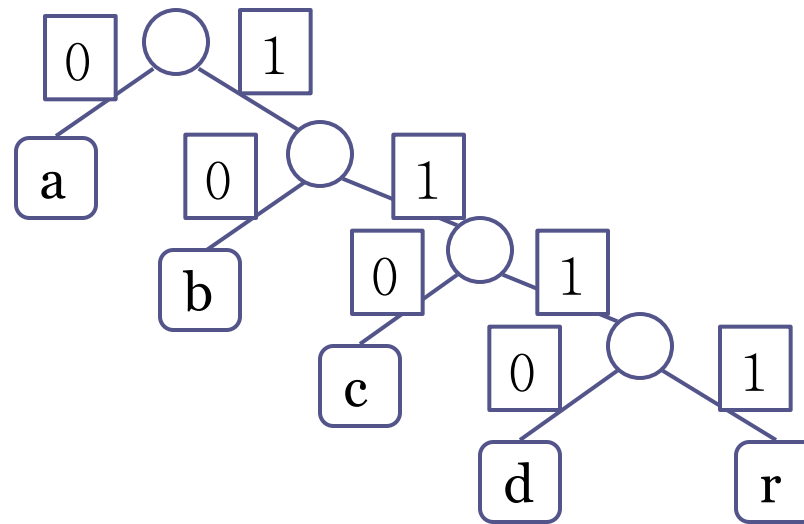
Exemplu decodificare



Input = -10111101100111001011110

Output = a-----

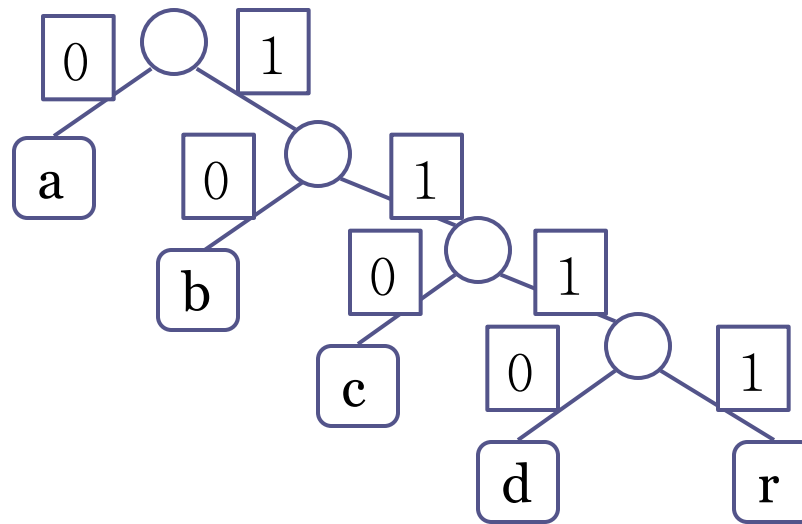
Exemplu decodificare



Input = -10111101100111001011110

Output = a-----

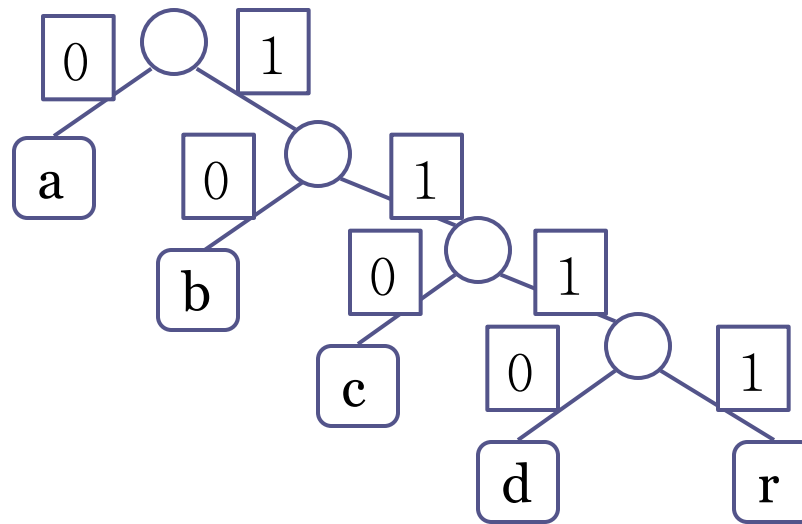
Exemplu decodificare



Input = --0111101100111001011110

Output = ab-----

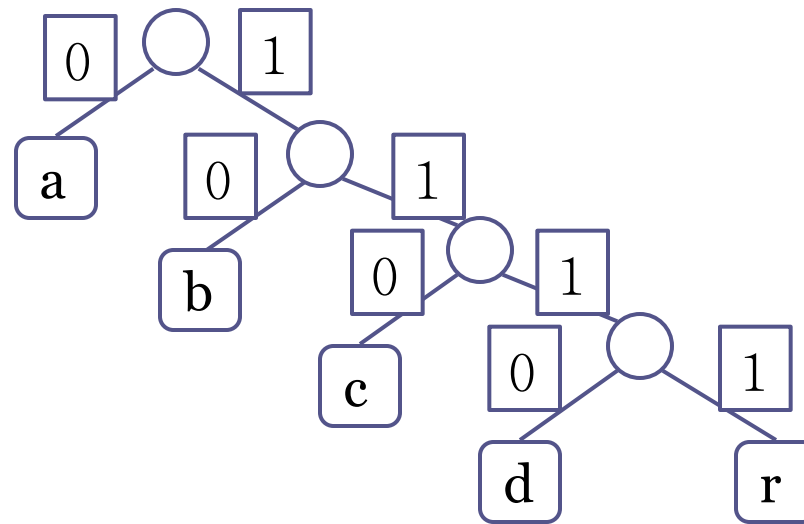
Exemplu decodificare



Input = ---111101100111001011110

Output = ab-----

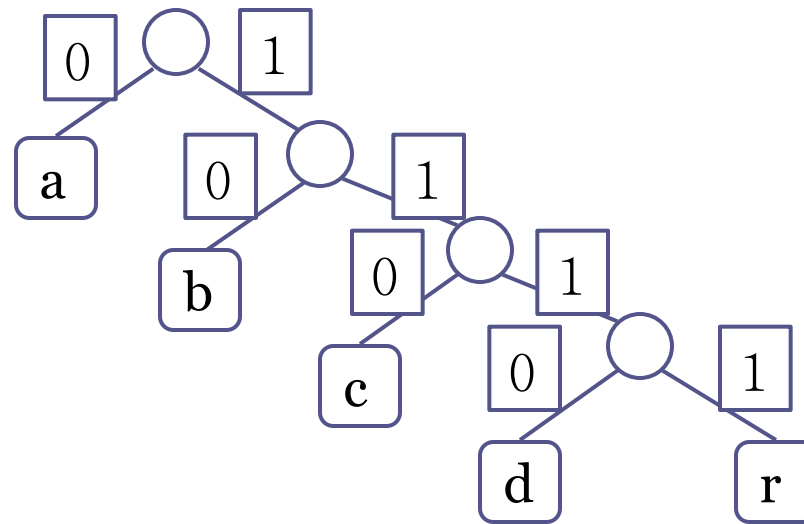
Exemplu decodificare



Input = ---111101100111001011110

Output = ab-----

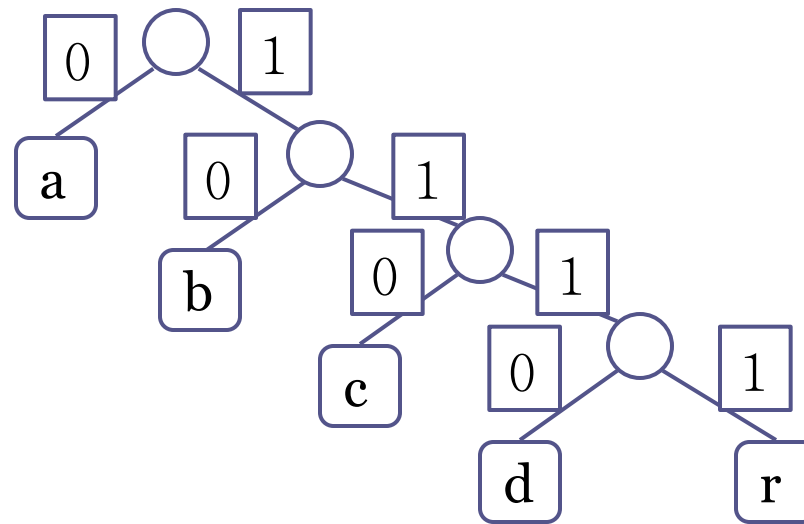
Exemplu decodificare



Input = ----11101100111001011110

Output = ab-----

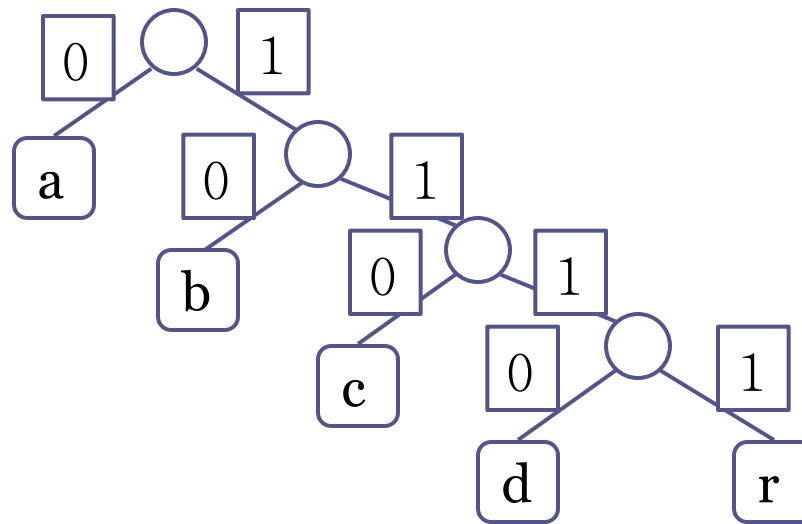
Exemplu decodificare



Input = -----1101100111001011110

Output = ab-----

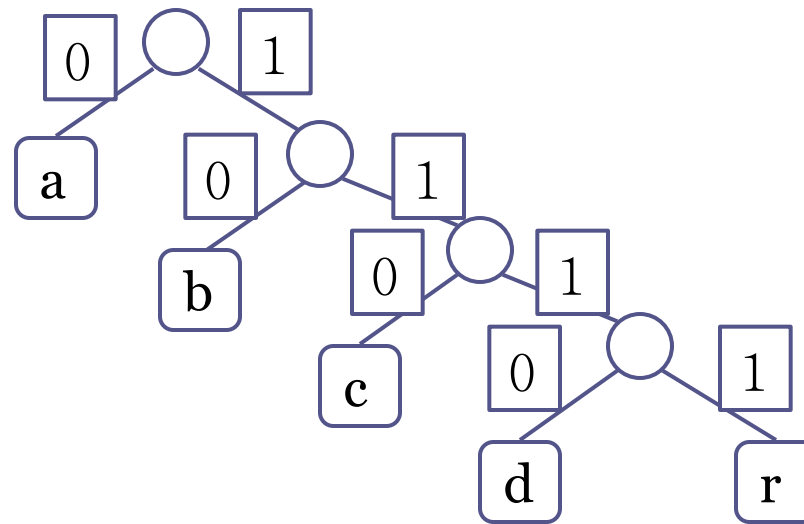
Exemplu decodificare



Input = -----101100111001011110

Output = abr-----

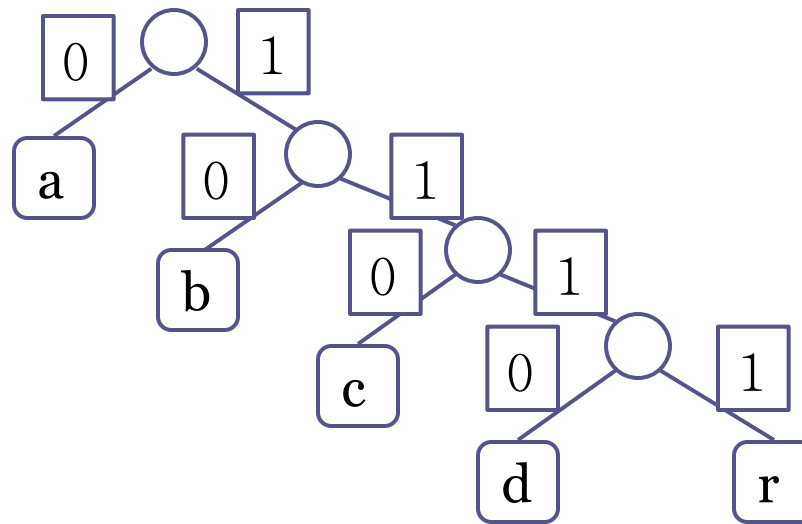
Exemplu decodificare



Input = -----01100111001011110

Output = abr-----

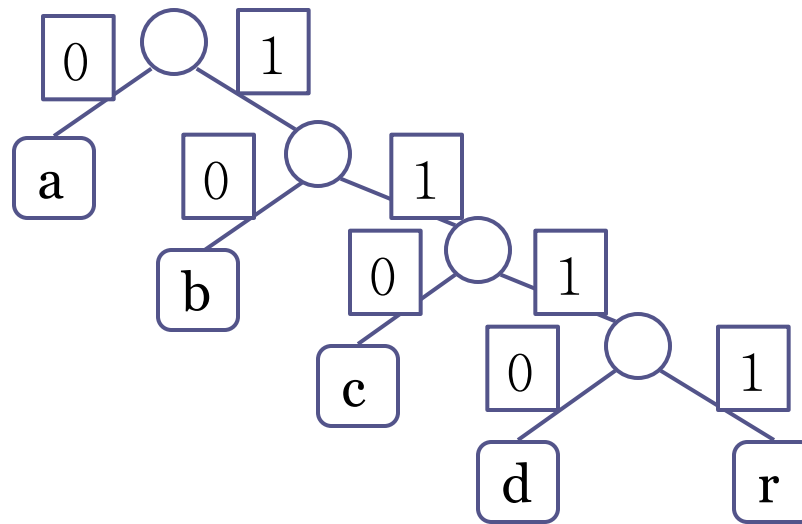
Exemplu decodificare



Input = -----01100111001011110

Output = abraa-----

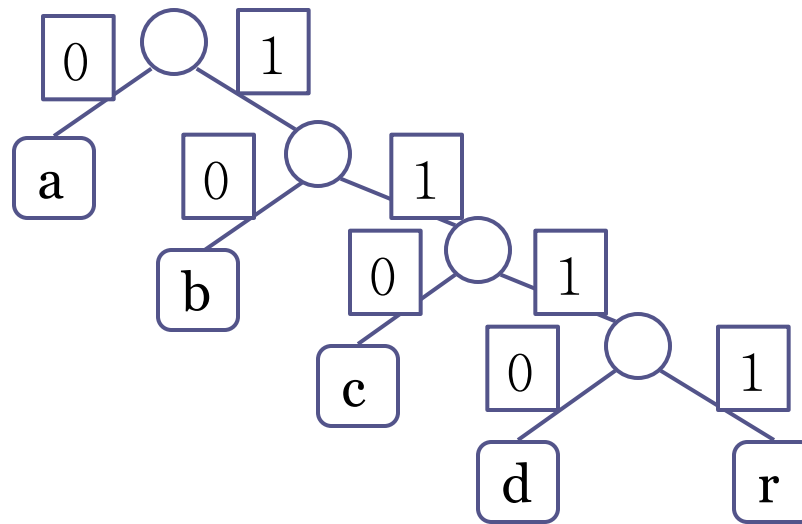
Exemplu decodificare



Input = -----1100111001011110

Output = abra-----

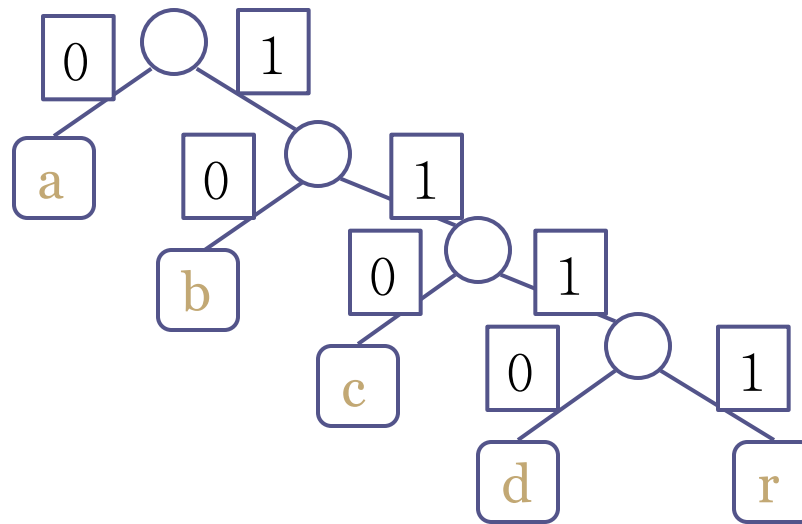
Exemplu decodificare



Input = -----1100111001011110

Output = abra-----

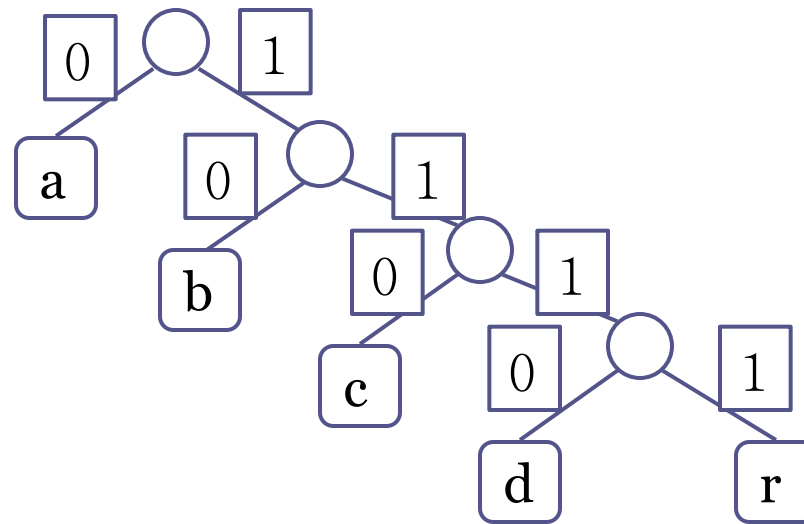
Exemplu decodificare



Input = -----100111001011110

Output = abra-----

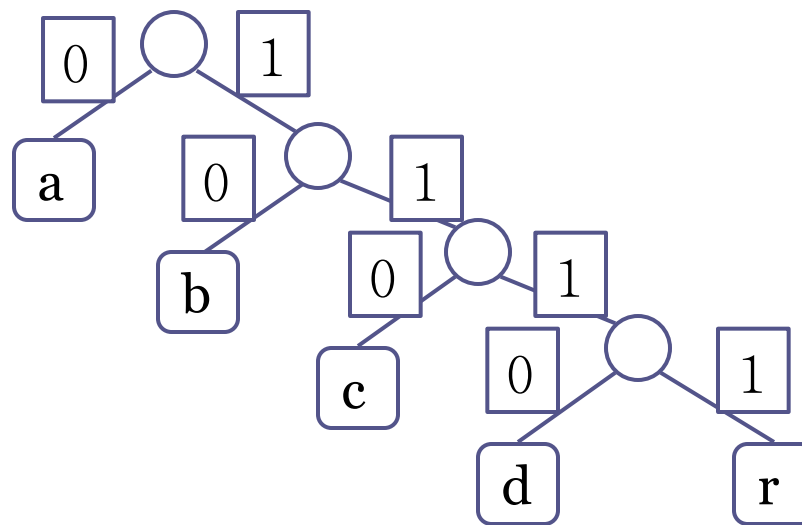
Exemplu decodificare



Input = -----00111001011110

Output = abrac-----

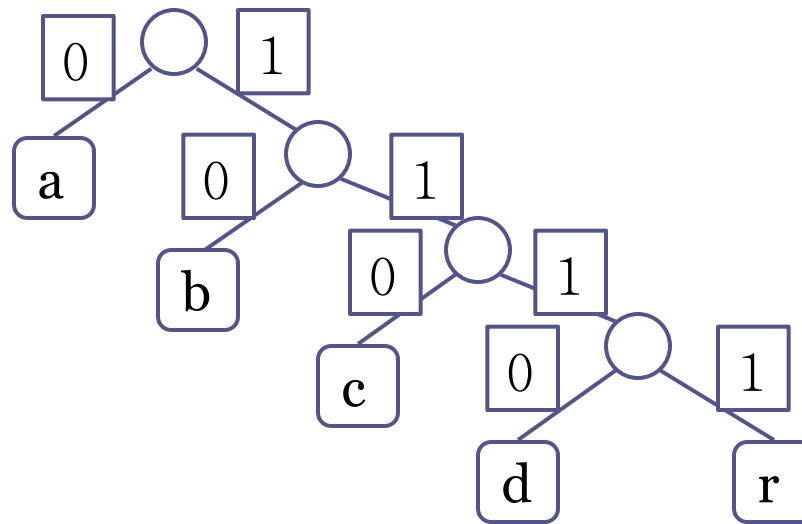
Exemplu decodificare



Input = -----0111001011110

Output = abra-----

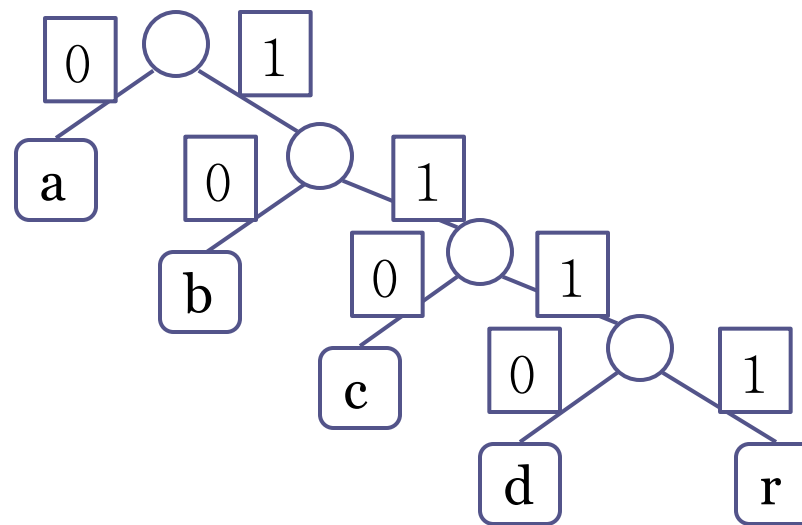
Exemplu decodificare



Input = -----0111001011110

Output = abraca-----

Exemplu decodificare



Si tot asa...

Input = -----111001011110

Output = abra-----