Αριθμητική Κωδικοποίηση

- Ο κώδικας Huffmann είναι βέλτιστος γιατί παράγει συμπαγή κώδικα για δεδομένες πιθανότητες
  - Συμπαγής κώδικας: Δεν υπάρχει άλλος με μικρότερο μέσο μήκος κωδικής λέξης
- Δεν είναι 100% αποτελεσματικός αν τα $\log(p_i)$ δεν είναι ακέραια μήκη.
  - Απαιτείται στρογγυλοποίηση προς τα επάνω ή ν-οστή επέκταση της πηγής για να προσεγγιστεί η εντροπία.
- Έχει επινοηθεί κωδικοποίηση που στρογγυλοποιεί προς τα κάτω.
- Δε χρειάζεται ούτε λεξικό!
  - Ο κώδικας παράγεται on-line

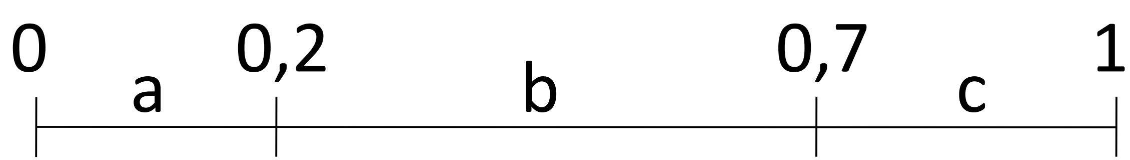
Αριθμητική Κωδικοποίηση

Σύμβολο	Πιθανότητα
a	0,2
b	0,5
c	0,3

Αντιστοίχιση κάθε συμβόλου σε τμήμα του διαστήματος [0, 1] ανάλογα με την πιθανότητα εμφάνισής του.

Αναπαράσταση του συμβόλου με έναν αριθμό από το διάστημά του και απεικόνιση του αρχικού διαστήματος στο διάστημα του συμβόλου

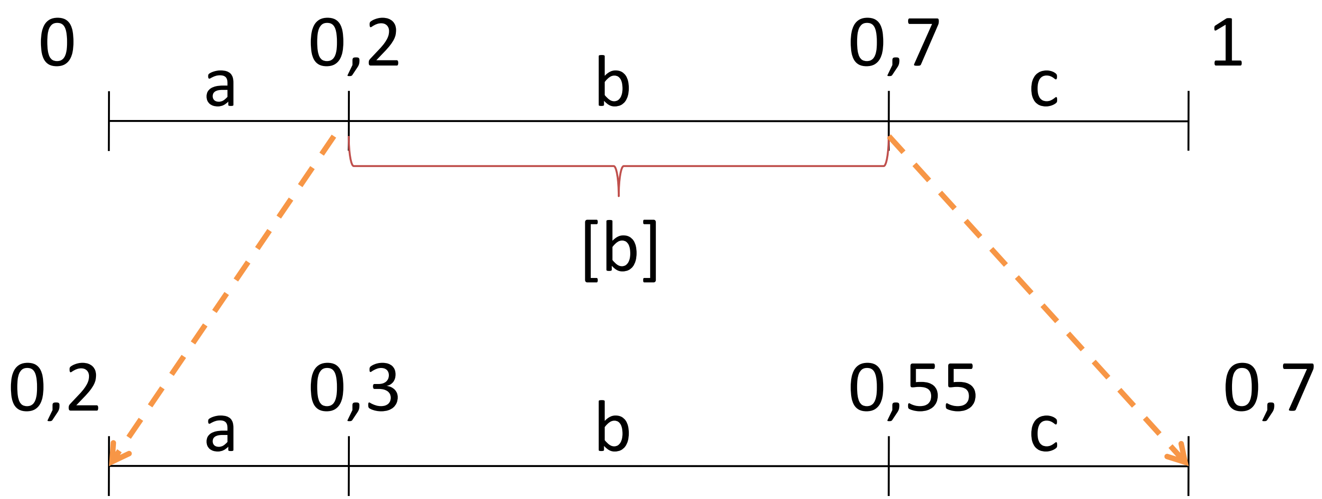
Κάθε επόμενο σύμβολο απεικονίζει το τρέχον διάστημα στο δικό του τμήμα του διαστήματος.



Αριθμητική Κωδικοποίηση

Έστω το μήνυμα [babca]. Πρώτο σύμβολο είναι το «b».

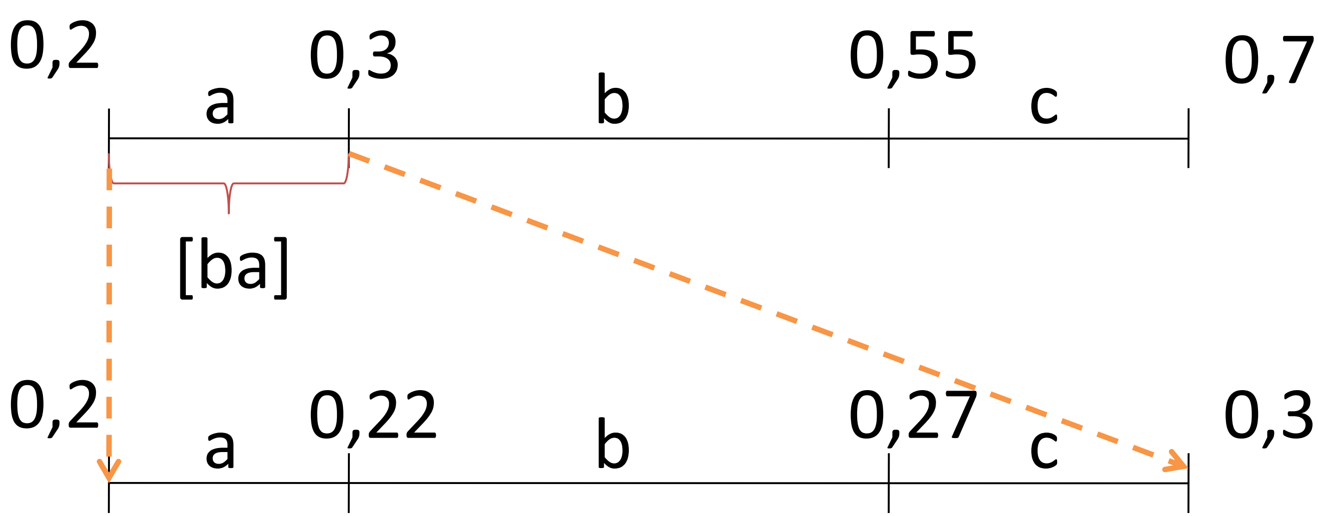
Χωρίζουμε το αντίστοιχο διάστημα [0.2, 0.7] σε τμήματα ανάλογα των πιθανοτήτων των συμβόλων της πηγής.



Αριθμητική Κωδικοποίηση

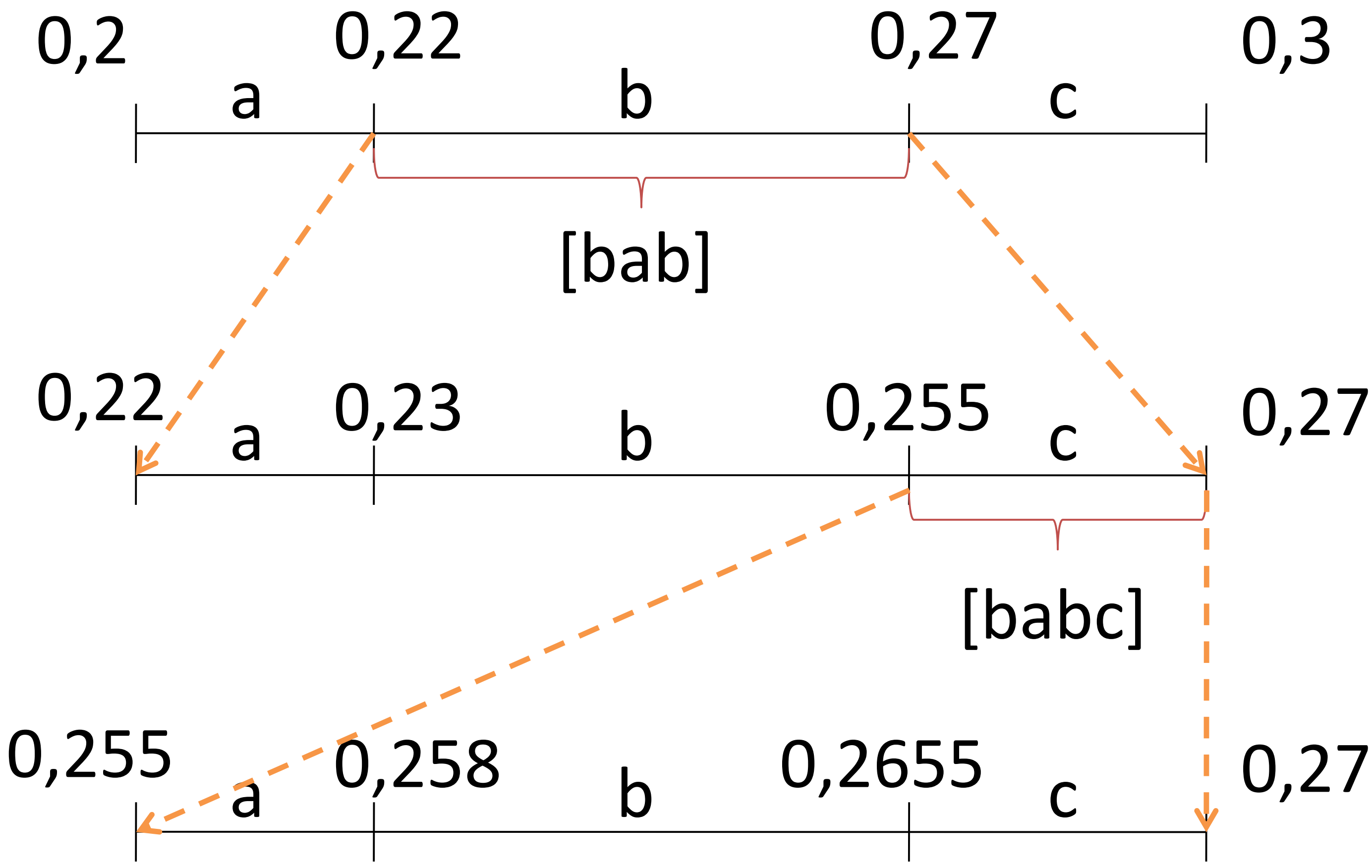
Το δεύτερο σύμβολο του μηνύματος [babca] είναι το «a».

Χωρίζουμε το αντίστοιχο διάστημα [0.2, 0.3] σε τμήματα ανάλογα των πιθανοτήτων των συμβόλων της πηγής.



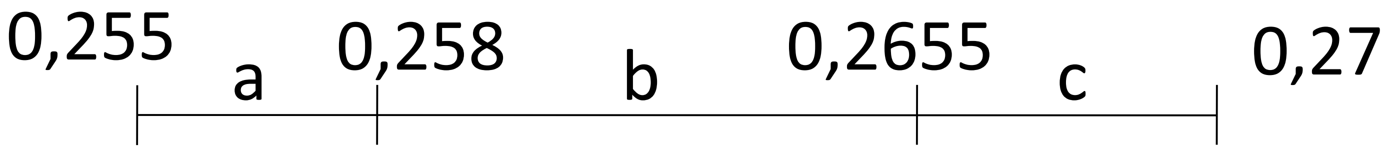
Αριθμητική Κωδικοποίηση

Τα υπόλοιπα βήματα είναι:



Αριθμητική Κωδικοποίηση

Έχουμε καταλήξει στην απεικόνιση:

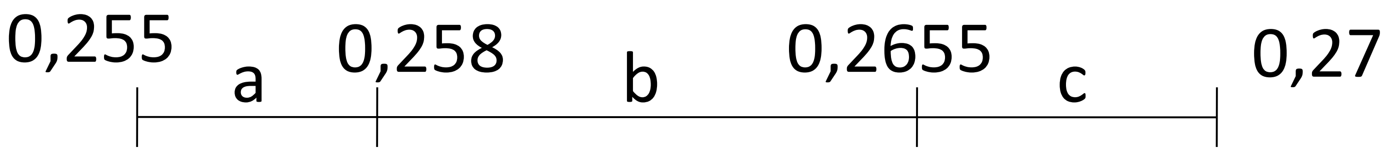


Τελευταίο σύμβολο στο μήνυμα είναι το «a».

Επιλέγεται ένας αριθμός από το διάστημα [0.255, 0.258] για την αναπαράσταση του μηνύματος [babca].

Έστω ο 0,257 που σε δυαδική μορφή είναι
$(0,257)_{10}=(0,010000011100...)_2$

Αριθμητική Κωδικοποίηση



Ο αριθμός bits που απαιτείται για την αναπαράσταση του μηνύματος προκύπτει από τη σχέση

$$k = \lceil -\log(\Delta) \rceil + 1$$

όπου Δ είναι το εύρος του τελικού διαστήματος:

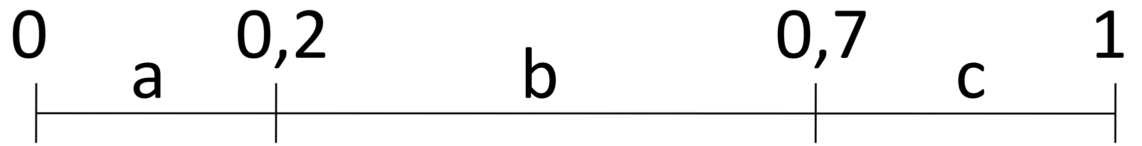
$$k = \lceil -\log(0.258 - 0.255) \rceil + 1 = 10 \quad \text{bits}$$

Τελικά, το μήνυμα [babca] κωδικοποιείται με την ακολουθία δυαδικών ψηφίων 0100000111.

Αλγόριθμος αριθμητικής κωδικοποίησης

- Low= 0;High=1
- While (not end of message)
  - Read(symbol x, low limit lx, probability px)
  - Range = High-Low
  - Low = Low+Range*lx
  - High = Low+Range*px
- end

Αριθμητική Αποκωδικοποίηση



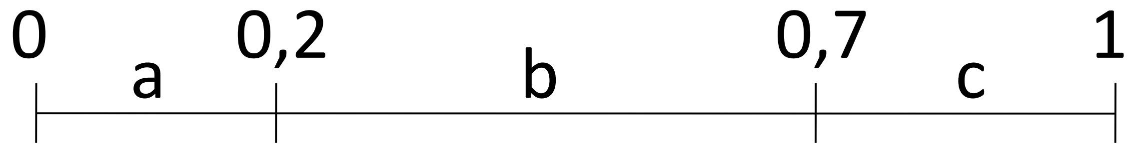
Ο δέκτης λαμβάνει το δυαδικό αριθμό d=0.0100000111 και το μετατρέπει σε δεκαδικό:

$$d=(0,010000011100)_2=0,2568359375$$

Ο αριθμός d ανήκει στο διάστημα [0.2, 0.7] και αποκωδικοποιείται ως «b».

Στη συνέχεια αφαιρείται η «επίδραση» του «b» από τον αριθμό d και συνεχίζουμε την αποκωδικοποίηση.

Αριθμητική Αποκωδικοποίηση

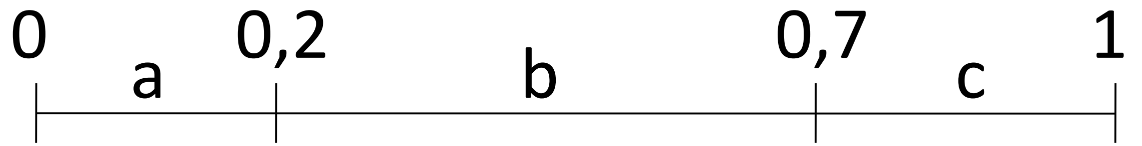


d = (d - low_b)/p_b =
(0,2568359375 - 0,2) / 0,5 = 0,113671875

Ο νέος αριθμός d ανήκει στο διάστημα [0, 0.2] και αποκωδικοποιείται ως «a». Συνολικά έχουμε «ba».

Στη συνέχεια αφαιρείται η «επίδραση» του «a» από τον αριθμό d και συνεχίζουμε την αποκωδικοποίηση.

Αριθμητική Αποκωδικοποίηση

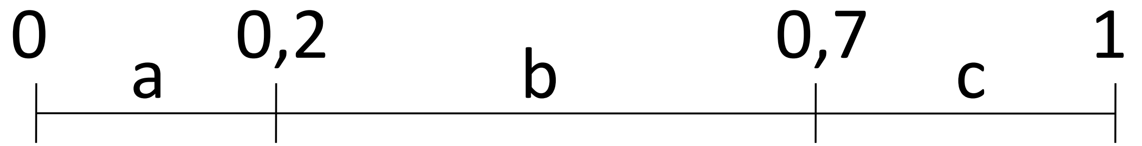


d = (d – low_a)/p_a =
(0,113671875 - 0) / 0,2 = 0,568359375

Ο νέος αριθμός d ανήκει στο διάστημα [0.2, 0.7] και αποκωδικοποιείται ως «b». Συνολικά έχουμε «bab».

Στη συνέχεια αφαιρείται η «επίδραση» του «b» από τον αριθμό d και συνεχίζουμε την αποκωδικοποίηση.

Αριθμητική Αποκωδικοποίηση

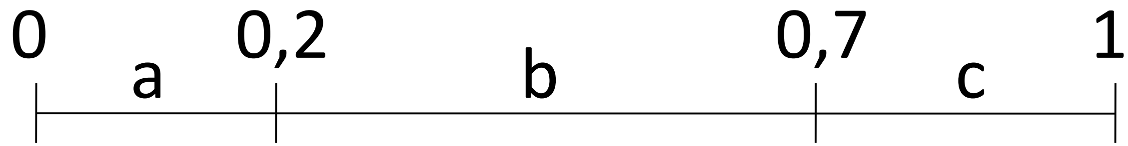


d = (d – low_b)/p_b =
(0,568359375 – 0,2) / 0,5 = 0,73671875

Ο νέος αριθμός d ανήκει στο διάστημα [0.7, 1] και αποκωδικοποιείται ως «c». Συνολικά έχουμε «babc».

Στη συνέχεια αφαιρείται η «επίδραση» του «c» από τον αριθμό d και συνεχίζουμε την αποκωδικοποίηση.

Αριθμητική Αποκωδικοποίηση



d = (d – low_c)/p_c =
(0,73671875– 0,3) / 0,5 = 0,1223958333333

Ο νέος αριθμός d ανήκει στο διάστημα [0, 0.2] και αποκωδικοποιείται ως «a». Συνολικά έχουμε «babca».

Εδώ πρέπει με κάποιο τρόπο να σημειώνεται το τέλος του μηνύματος. Διαφορετικά, η διαδικασία συνεχίζεται επ’άπειρον.

Αλγόριθμος αριθμητικής αποκωδικοποίησης

- Get(d,n)

/* n=number of symbols to decode */

- For k=1:n
  - x = decode(d, low_x, high_x)
  - d = d - low_x
  - d=d/p_x
- end

Huffmann vs Αριθμητικής Κωδικοποίησης

- Το 1ο θεώρημα του Shannon λέει ότι ο κώδικας Huffmann για τη ν-οστή επέκταση της πηγής δίνει μέσο μήκος κωδικής λέξης το πολύ $H(A)+1/\nu$.
- Έχει αποδειχτεί ότι η αριθμητική κωδικοποίηση μηνύματος μήκους ν δίνει μέσο μήκος κωδικής λέξης το πολύ $H(A)+1/\nu$.
  - Συμπαγής κώδικας: Δεν υπάρχει άλλος με μικρότερο μέσο μήκος κωδικής λέξης
- Θεωρητικά έχουν την ίδια απόδοση
- Ο κώδικας Huffmann έχει μεγαλύτερη πολυπλοκότητα για ν-οστές επεκτάσεις πηγών.