

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Τεχνολογία Πολυμέσων

Ενότητα # 9: Κωδικοποίηση εντροπίας

Διδάσκων: Γεώργιος Ξυλωμένος

Τμήμα: Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Χρηματοδότηση

- Το παρόν εκπαιδευτικό υλικό έχει αναπτυχθεί στα πλαίσια του εκπαιδευτικού έργου του διδάσκοντα.
- Το έργο «**Ανοικτά Ακαδημαϊκά Μαθήματα στο Οικονομικό Πανεπιστήμιο Αθηνών**» έχει χρηματοδοτήσει μόνο τη αναδιαμόρφωση του εκπαιδευτικού υλικού.
- Το έργο υλοποιείται στο πλαίσιο του Επιχειρησιακού Προγράμματος «Εκπαίδευση και Δια Βίου Μάθηση» και συγχρηματοδοτείται από την Ευρωπαϊκή Ένωση (Ευρωπαϊκό Κοινωνικό Ταμείο) και από εθνικούς πόρους.



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



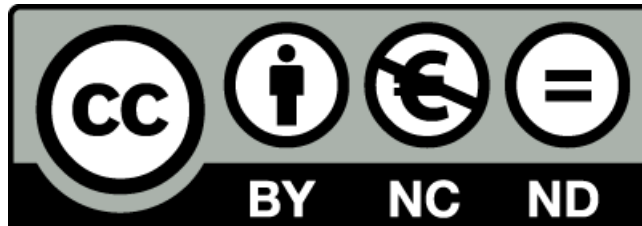
ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Άδειες Χρήσης

- Το παρόν εκπαιδευτικό υλικό υπόκειται σε άδειες χρήσης Creative Commons.
- Οι εικόνες προέρχονται από το βιβλίο «Τεχνολογία Πολυμέσων και Πολυμεσικές Επικοινωνίες», Γ.Β. Ξυλωμένος, Γ.Κ. Πολύζος, 1<sup>η</sup> έκδοση, 2009, Εκδόσεις Κλειδάριθμος.



# Σκοποί ενότητας

- Κατανόηση των βασικών τεχνικών κωδικοποίησης εντροπίας με χρήση δένδρου (Shannon-Fano και Huffman).
- Εξοικείωση με την τεχνική της αριθμητικής κωδικοποίησης.
- Εισαγωγή στις τεχνικές κωδικοποίησης με παράθυρο (LZ77/LZSS) και με λεξικό (LZ78/LZW).

# Περιεχόμενα ενότητας

- Βέλτιστη κωδικοποίηση
- Κωδικοποίηση Shannon-Fano
- Κωδικοποίηση Huffman
- Προσαρμοστική Huffman
- Αριθμητική κωδικοποίηση
- Κωδικοποίηση με παράθυρο
- Κωδικοποίηση με λεξικό

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

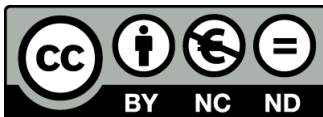


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Βέλτιστη κωδικοποίηση

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Βέλτιστη κωδικοποίηση (1/4)

- Κωδικοποίηση σταθερού μήκους
  - Παράδειγμα: ASCII
  - Κάθε χαρακτήρας είναι 7 bits
- Κωδικοποίηση μεταβλητού μήκους (VLC)
  - Παράδειγμα: κώδικας Morse
  - Τρία σύμβολα (dot/dash/space)
  - Περισσότερα σύμβολα σε σπάνιους χαρακτήρες
  - Χρήση διαστημάτων για διαχωρισμό

# Βέλτιστη κωδικοποίηση (2/4)

## International Morse Code

1. The length of a dot is one unit.
2. A dash is three units.
3. The space between parts of the same letter is one unit.
4. The space between letters is three units.
5. The space between words is seven units.

A	• —	U	• • —
B	— • • •	V	• • • —
C	— • — •	W	• — —
D	— • •	X	— • • —
E	•	Y	— • — —
F	• • — •	Z	— — • •
G	— — •		
H	• • • •		
I	• •		
J	• — — —		
K	— • —	1	• — — — —
L	• — • •	2	• • — — —
M	— —	3	• • • — —
N	— •	4	• • • • —
O	— — —	5	• • • • •
P	• — — •	6	— • • • •
Q	— — • —	7	— — • • •
R	• — •	8	— — — • •
S	• • •	9	— — — — •
T	—	0	— — — — —



# Βέλτιστη κωδικοποίηση (3/4)

- Βέλτιστη κωδικοποίηση εντροπίας
  - Τόσα bits όσα η εσωτερική πληροφορία
    - Μέσο μήκος = εντροπία πηγής
  - Αν οι πληροφορίες δεν είναι ακέραιοι αριθμοί;
    - Χάνουμε λίγο σε απόδοση
- Χρήση μόνο 0 και 1 (χωρίς διαχωριστικό)
  - Πώς διαχωρίζουμε τους κωδικούς;
  - Με την ιδιότητα μοναδικού προθέματος
    - Κανένα σύμβολο δεν είναι πρόθεμα άλλου

# Βέλτιστη κωδικοποίηση (4/4)

- Απαιτείται γνώση των πιθανοτήτων εισόδου
  - Είτε διαβάζουμε πρώτα ολόκληρο το αρχείο
    - Αν δεν έχουμε όλα τα δεδομένα διαθέσιμα;
  - Είτε υποθέτουμε κάποια κατανομή
  - Είτε υπολογίζουμε σταδιακά την κατανομή
- Άρα, βέλτιστή υπό προϋποθέσεις!
  - Υπάρχουν τέτοιες κωδικοποιήσεις;
  - Ναι! Shannon-Fano και Huffman

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κωδικοποίηση Shannon-Fano

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



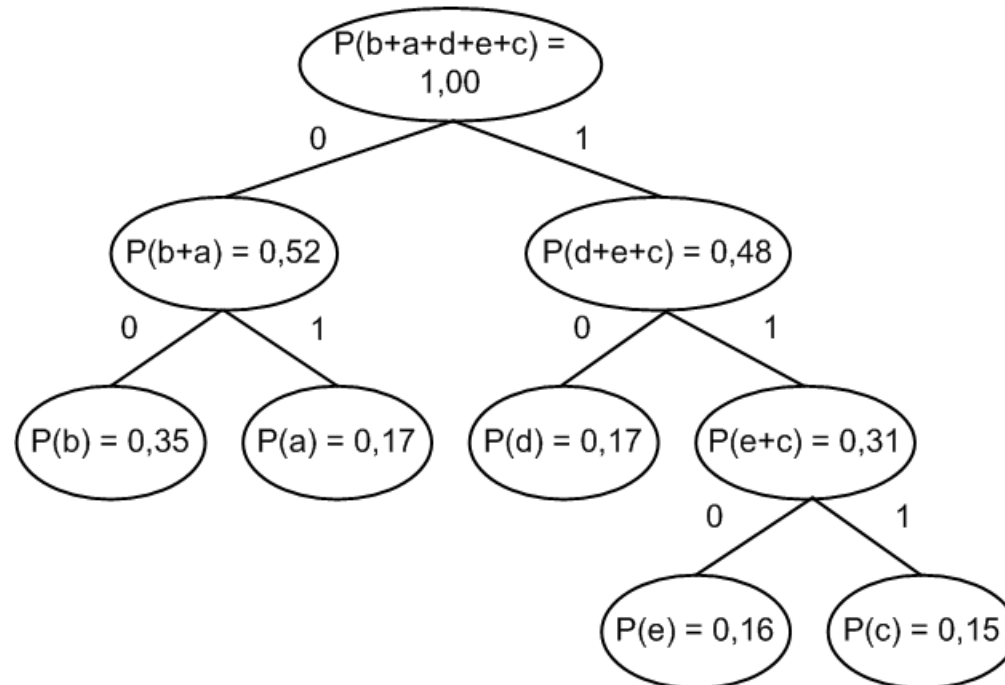
# Μέθοδος Shannon-Fano (1 από 5)

- Κωδικοποίηση Shannon-Fano
  - Χρήση κωδικών (εξόδου) με ακέραιο μήκος bit
    - Απόκλιση από ιδανική συμπίεση
  - Κανένα κωδικός δεν είναι πρόθεμα άλλου
    - Επιτρέπει αποκωδικοποίηση μεταβλητού μήκους
  - Δυαδικό δένδρο κωδικοποίησης
    - Φύλλα: σύμβολα και πιθανότητες
    - Κόμβοι: σύνολα συμβόλων και πιθανοτήτων
  - Ίδιο δένδρο αποκωδικοποίησης

# Μέθοδος Shannon-Fano (2 από 5)

- Κατασκευή δένδρου
  - Ταξινομούμε σύμφωνα με τις πιθανότητες
    - Αύξουσα ή φθίνουσα σειρά
  - Διάσπαση συμβόλων σε «ισοπίθανες» ομάδες
    - Κάθε ομάδα έχει το άθροισμα των πιθανοτήτων
    - Ομάδες με ελάχιστη διαφορά αθροίσματος
    - **Προσοχή:** τα σύμβολα δεν ταξινομούνται ξανά!
    - Οι δύο ομάδες γίνονται παιδιά του κόμβου
    - Αντιστοίχιση 0 και 1 στα δύο παιδιά
  - Επανάληψη μέχρι να μείνουν μόνο τα φύλλα
    - Κάθε φύλλο είναι ένα σύμβολο

# Μέθοδος Shannon-Fano (3 από 5)



- Παράδειγμα δένδρου κωδικοποίησης
  - $P(a)=0,17$ ,  $P(b)=0,35$ ,  $P(c)=0,15$ ,  $P(d)=0,17$ ,  $P(e)=0,16$
  - Η ταξινόμηση δίνει τη σειρά b, a, d, e, c
  - Μέσο μήκος κώδικα: 2,31

# Μέθοδος Shannon-Fano (4 από 5)

- Κωδικοποίηση: Αντικατάσταση  $x$  με  $w(x)$ 
  - Κάθε σύμβολο αντιστοιχεί σε ένα φύλλο
  - Το μονοπάτι του φύλλου είναι ο κωδικός
- Αποκωδικοποίηση
  - Απαιτείται γνώση δένδρου κωδικοποίησης
  - Αντιστοίχιση εισόδου με μονοπάτια δένδρου
    - Κάθε πρόθεμα αντιστοιχεί σε διαφορετικό μονοπάτι
    - Ξέρουμε πάντα πότε να σταματήσουμε
- Ξεκινάμε πάντα από τη ρίζα

# Μέθοδος Shannon-Fano (5 από 5)

- Κατασκευή δένδρου κωδικοποίησης
  - Υπολογισμός πιθανοτήτων από αρχείο
  - Χρήση έτοιμων δένδρων
    - Ουσιαστικά έτοιμων κατανομών πιθανότητας
- Κατασκευή δένδρου αποκωδικοποίησης
  - Μετάδοση δένδρου κωδικοποίησης
  - Μετάδοση πιθανοτήτων
    - Απαιτείται γνώση των κανόνων κωδικοποίησης
  - Χρήση έτοιμων δένδρων (πρέπει να συμφωνούν!)



**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κωδικοποίηση Huffman

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

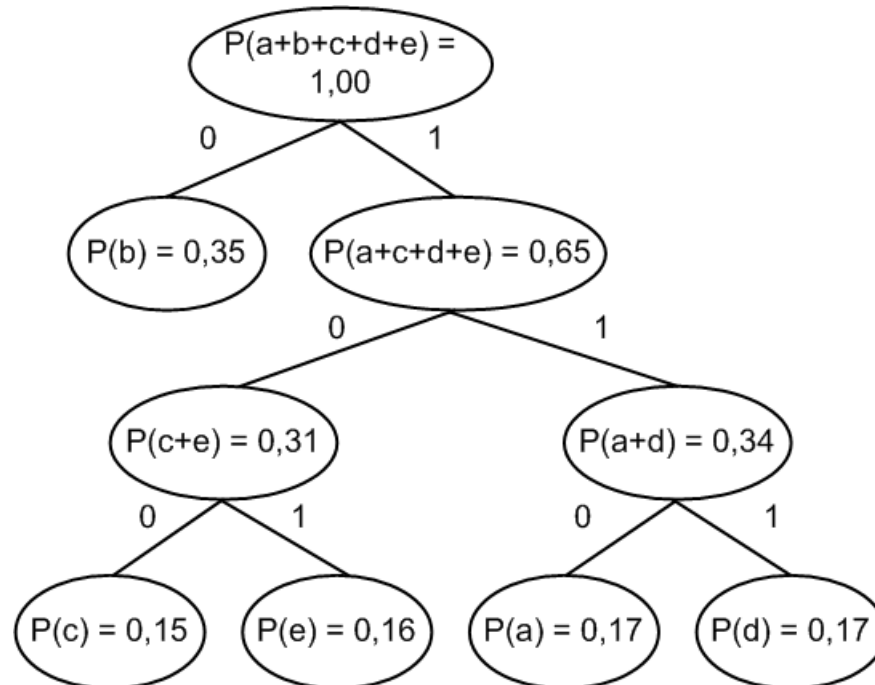
# Μέθοδος Huffman (1 από 3)

- Παρόμοια με Shannon-Fano
  - Μεταβλητό πλήθος bit ανά σύμβολο
  - Απαιτεί γνώση πιθανοτήτων εισόδου
  - Δυαδικό δένδρο (απο)κωδικοποίησης
    - Μπορεί να διαφέρει από Shannon-Fano
  - Ίδιος αλγόριθμος (από)κωδικοποίησης
  - Δημιουργία δένδρου με ανάποδο τρόπο

# Μέθοδος Huffman (2 από 3)

- Κατασκευή
  - Αρχικά κάθε σύμβολο είναι ένας κόμβος
  - Επιλογή κόμβων με ελάχιστες πιθανότητες
  - Αντικατάσταση κόμβων με υποδένδρο
    - Άθροισμα πιθανοτήτων παιδιών στον πατέρα
    - Αντιστοίχιση 0 και 1 στα δύο παιδιά
  - Σταματάμε όταν μείνει ένα δένδρο
    - Η ρίζα αντιπροσωπεύει όλα τα σύμβολα
  - Πιο ισορροπημένα δένδρα από Shannon-Fano

# Μέθοδος Huffman (3 από 3)



- Παράδειγμα δένδρου κωδικοποίησης
  - $P(a)=0,17$ ,  $P(b)=0,35$ ,  $P(c)=0,15$ ,  $P(d)=0,17$ ,  $P(e)=0,16$
  - Μέσο μήκος κώδικα: 2,3 (καλύτερο από Shannon-Fano)

# Huffman ή Shannon-Fano (1 από 3)

- Huffman ή Shannon-Fano;
  - Σχεδόν πανομοιότυποι αλγόριθμοι
  - Αλλάζει μόνο το δένδρο
  - Ίδια κωδικοποίηση/αποκωδικοποίηση
- Πιο απλή κατασκευή στο Shannon-Fano
  - ΔΕΝ ταξινομούμε τα σύμβολα σε κάθε βήμα
  - Εύκολος εντοπισμός του σημείου διάσπασης
    - Προσθέτω πιθανότητες από αριστερά προς τα δεξιά
    - Μόλις περάσω το μισό του συνόλου, επιλέγω

# Huffman ή Shannon-Fano (2 από 3)

- Πιο αποδοτική κωδικοποίηση στο Huffman
  - Στο Shannon-Fano δεν έχω βέλτιστο τεμαχισμό
    - Αλλάζοντας τη σειρά μπορεί να βρω καλύτερο
  - Στον Huffman ταξινομώ κάθε φορά το σύνολο
    - Θέλουμε πάντα το ελάχιστο υποδένδρο/φύλλο
  - Τι δομή δεδομένων μας συμφέρει;
    - Σε κάθε βήμα βγάζω δύο κόμβους και προσθέτω έναν
    - Δεν χρειάζεται πλήρης ταξινόμηση του συνόλου

# Huffman ή Shannon-Fano (3 από 3)

- Μειονεκτήματα Huffman/Shannon-Fano
  - Απαιτείται γνώση των πιθανοτήτων εισόδου
  - Η κωδικοποίηση δεν είναι βέλτιστη
    - Ακέραιο πλήθος bit ανά σύμβολο
    - Αναγκαστικά αποκλίνουμε από το ιδανικό
  - Βελτίωση συμπίεσης
    - Κωδικοποίηση ομάδων από  $n$  σύμβολα;
    - Εκθετική αύξηση του πλήθους των συμβόλων

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Προσαρμοστική Huffman

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ



# Προσαρμοστική Huffman (1 από 11)

- Προσαρμοστική μέθοδος Huffman
  - Δεν χρειάζεται γνώση των συχνοτήτων
    - Το δένδρο χτίζεται σταδιακά
    - Και αλλάζει ανάλογα με την είσοδο
  - Ξεκινάμε με μία αρχική κωδικοποίηση
    - Μπορεί να είναι απλά τα 8 bit του Extended ASCII
    - Οι κωδικοί σταδιακά αλλάζουν
      - Ανάλογα με τη συχνότητα των συμβόλων

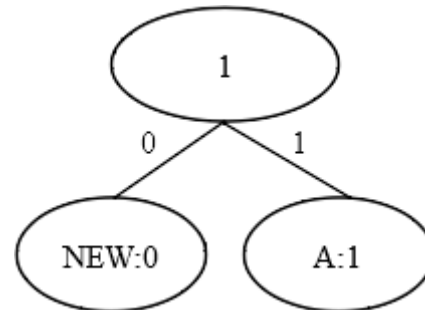
# Προσαρμοστική Huffman (2 από 11)

- Προσαρμοστική μέθοδος Huffman
  - Για κάθε σύμβολο κρατάμε συχνότητα
    - Αυξάνεται κάθε φορά που βλέπουμε το σύμβολο
    - Αρχικά όλες οι συχνότητες είναι 0
  - Το αρχικό δένδρο έχει μόνο τον κωδικό NEW
    - Έχει συχνότητα 0 (δεν είναι ποτέ σύμβολο εισόδου)
    - Χρησιμοποιείται όταν βλέπουμε νέο σύμβολο
    - Μας λέει ότι το νέο σύμβολο θα μπει στο δένδρο

# Προσαρμοστική Huffman (3 από 11)

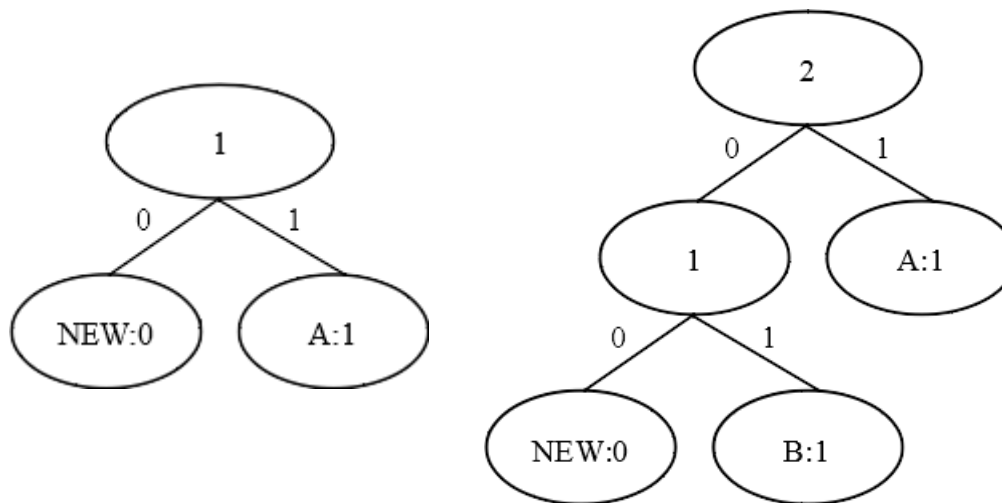
- Όταν βλέπουμε νέο σύμβολο
  - Πρώτα εξάγεται το NEW
  - Μετά εξάγεται ο αρχικός κωδικός
  - Τέλος, το σύμβολο εισάγεται στο δένδρο
  - Η αρχική συχνότητά του είναι 1
- Όταν βλέπουμε ένα υπάρχον σύμβολο
  - Εξάγουμε τον τρέχοντα κωδικό του
  - Αυξάνουμε τη συχνότητά του

# Προσαρμοστική Huffman (4 από 11)



- Παράδειγμα: συμβολοσειρά ABCCA
  - Αρχικοί κωδικοί: A=01, B=10, C=11
  - Αρχικό δένδρο: NEW=0
  - ABCCA: έξοδος 001 (NEW και A), το A στο δένδρο

# Προσαρμοστική Huffman (5 από 11)

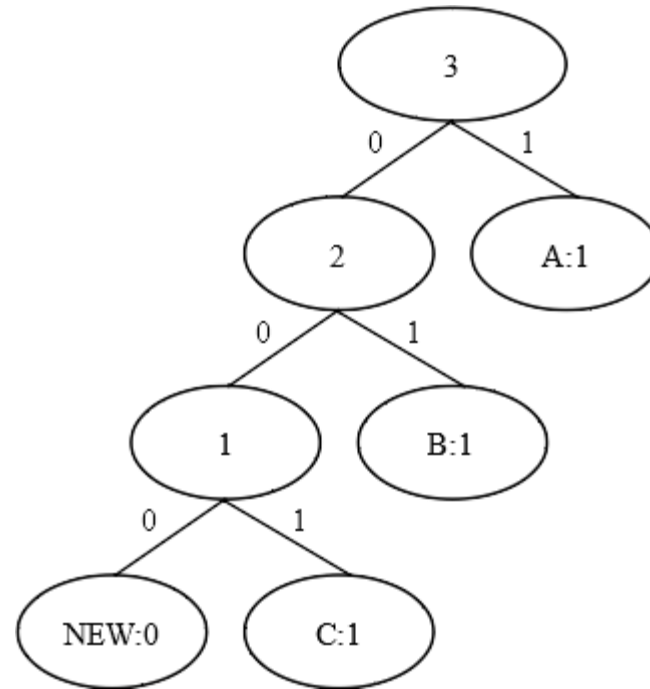
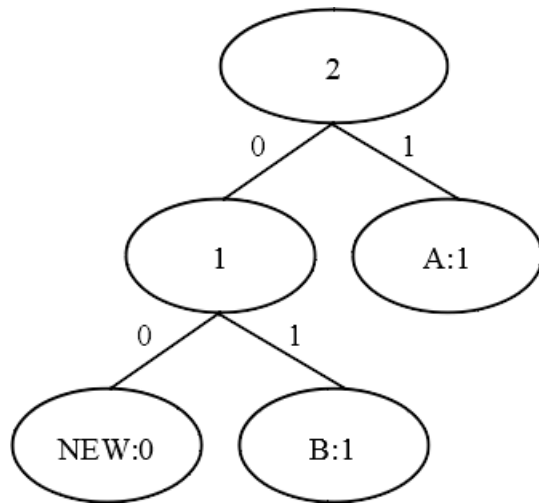


- Παράδειγμα: συμβολοσειρά ABCCA
  - Η ρίζα έχει το άθροισμα συχνοτήτων των παιδιών
  - ABCCA: έξοδος 010 (NEW και B), το B στο δένδρο
  - Ενημερώνονται και οι εσωτερικοί κόμβοι

# Προσαρμοστική Huffman (6 από 11)

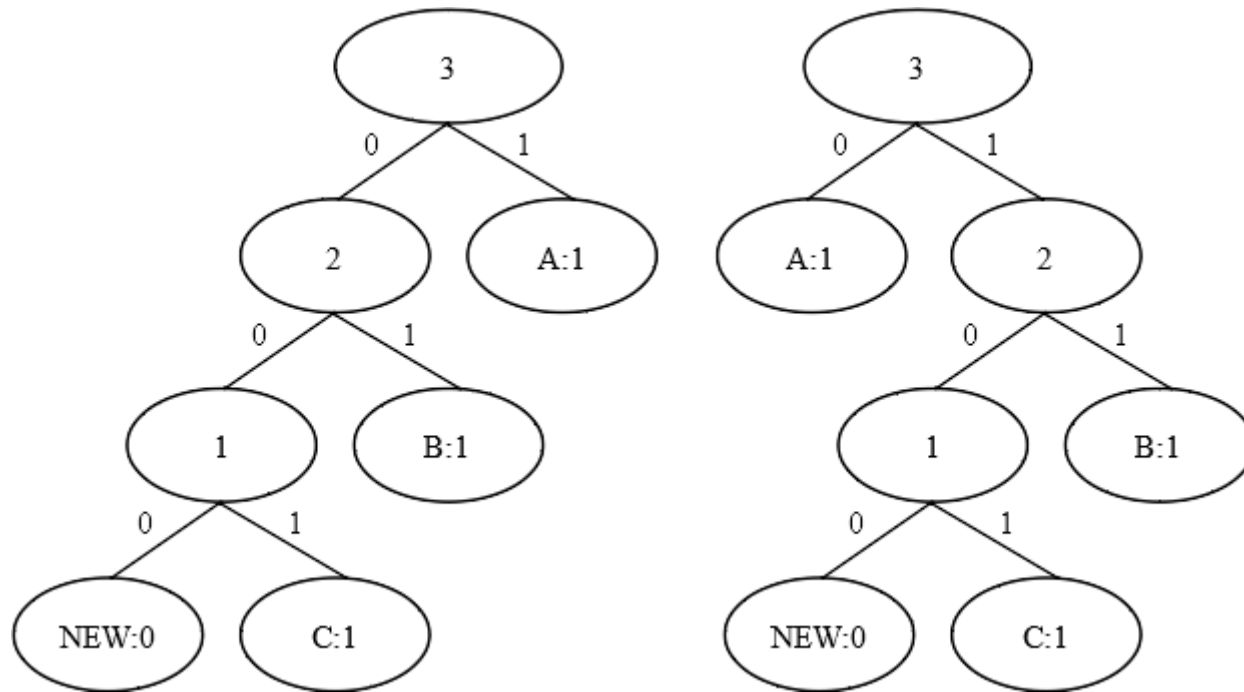
- Το δένδρο είναι πάντα ταξινομημένο
  - Ως προς τις συχνότητες
    - Κάτω προς πάνω, αριστερά προς δεξιά
  - Έστω ότι άλλαξε η συχνότητα από  $N$  σε  $N+1$ 
    - Αν ο κόμβος δεν είναι στη σωστή θέση
    - Βρίσκω τον πιο μακρινό κόμβο με  $N$
    - Αλλάζω τους δύο κόμβους (μπορεί να είναι δένδρα)
    - Συνεχίζω αναδρομικά όσο χρειάζεται
  - Ο παραλήπτης κάνει ακριβώς την ίδια δουλειά

# Προσαρμοστική Huffman (7 από 11)



- ABCCA: έξοδος 0011 (NEW και C), το C στο δένδρο
  - Το NEW έχει αλλάξει κωδικό στο προηγούμενο βήμα
- Στο πρώτο επίπεδο η σειρά είναι λάθος

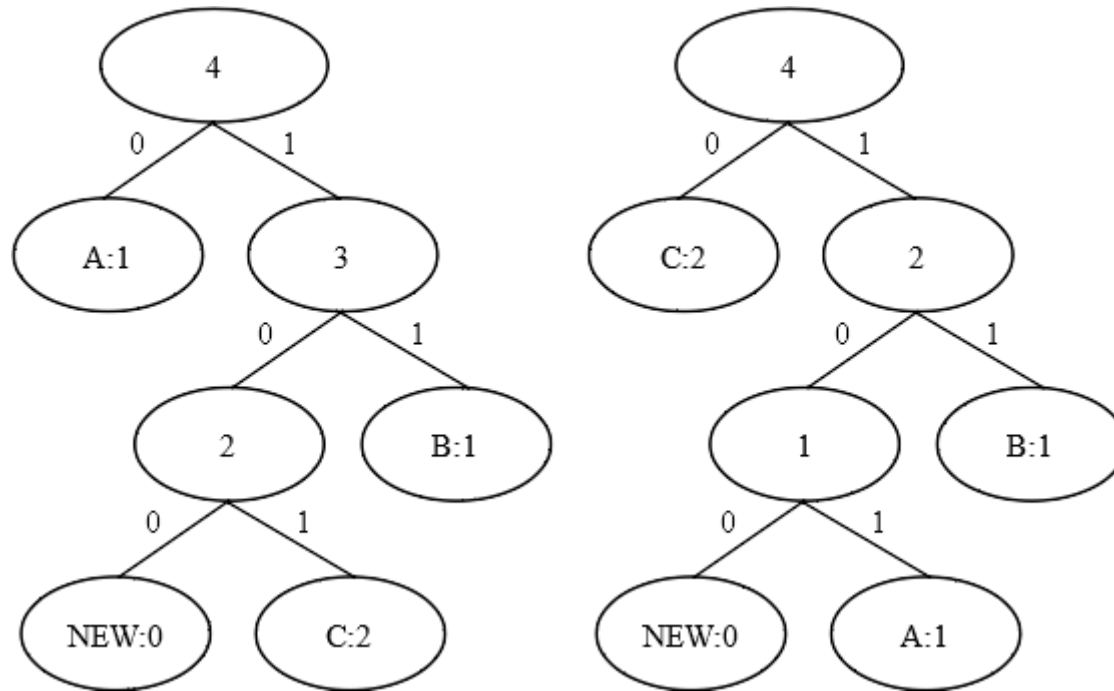
# Προσαρμοστική Huffman (8 από 11)



- Πρέπει να αλλάξει η σειρά στο πρώτο επίπεδο
- Αλλάζουμε το A με τον πιο μακρινό κόμβο με 2
  - Πρόκειται για ολόκληρο υποδένδρο



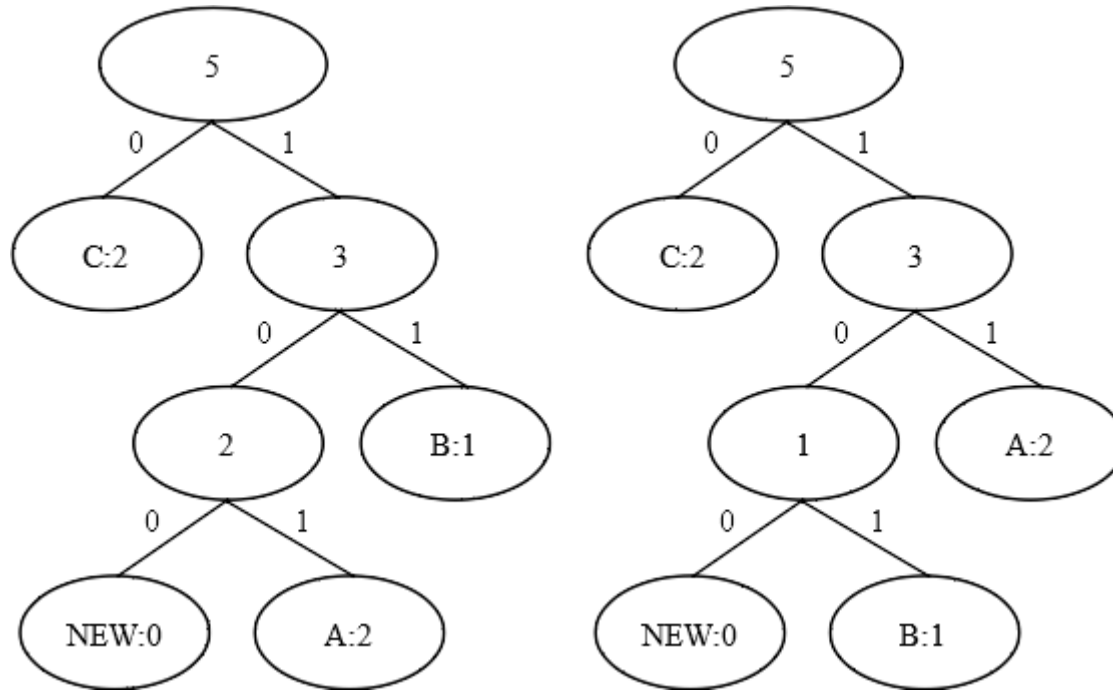
# Προσαρμοστική Huffman (9 από 11)



– ABCCA: έξοδος 101

– Το C αλλάζει με το A (το πιο μακρινό με 1)

# Προσαρμοστική Huffman (10 από 11)



- ABCCA: έξοδος 101
- Το A αλλάζει με το B

# Προσαρμοστική Huffman (11 από 11)

- Συγχρονισμός των δύο άκρων
  - Πρώτα εξάγουμε κωδικό
  - Μετά αλλάζουμε το δένδρο
    - Επιτρέπει στον αποκωδικοποιητή να ακολουθεί
    - Ξέρει ποιος χαρακτήρας αλλάζει συχνότητα
  - Πριν τους νέους κωδικούς εξάγουμε NEW
    - Ακολουθείται από αρχικό (προσωρινό) κωδικό
  - Παρατήρηση: το NEW αλλάζει κι αυτό κωδικό

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Αριθμητική κωδικοποίηση

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



# Αριθμητική (1 από 10)

- Κωδικοποίηση εισόδου με έναν αριθμό
  - Το πλήθος bit εξαρτάται από το μήκος
    - Μπορεί να είναι πάρα πολύ μεγάλο
  - Δεν απαιτείται σταθερό πλήθος bit/σύμβολο
    - Αποφεύγουμε την απόκλιση από το βέλτιστο
  - Απαιτείται γνώση των πιθανοτήτων εισόδου
  - Απαιτείται τερματικό σύμβολο στο τέλος
    - Για να ξέρουμε ότι τελείωσε η αποκωδικοποίηση

# Αριθμητική (2 από 10)

- Προετοιμασία αλγορίθμου
  - Ταξινόμηση συμβόλων (συνήθως αλφαβητικά)
  - Στο  $x_i$  αντιστοιχίζουμε το διάστημα  $[a_i, b_i)$ 
    - Το διάστημα ορίζεται έτσι ώστε  $b_i - a_i = p(x_i)$
  - Παράδειγμα
    - $P(a) = 0.4, P(b) = 0.3, P(c) = 0.2$  και  $P(\$) = 0.1$
    - Διάστημα a:  $[0, 0.4)$ , Διάστημα b:  $[0.4, 0.7)$
    - Διάστημα c:  $[0.7, 0.9)$ , Διάστημα \$:  $[0.9, 1.0)$

# Αριθμητική (3 από 10)

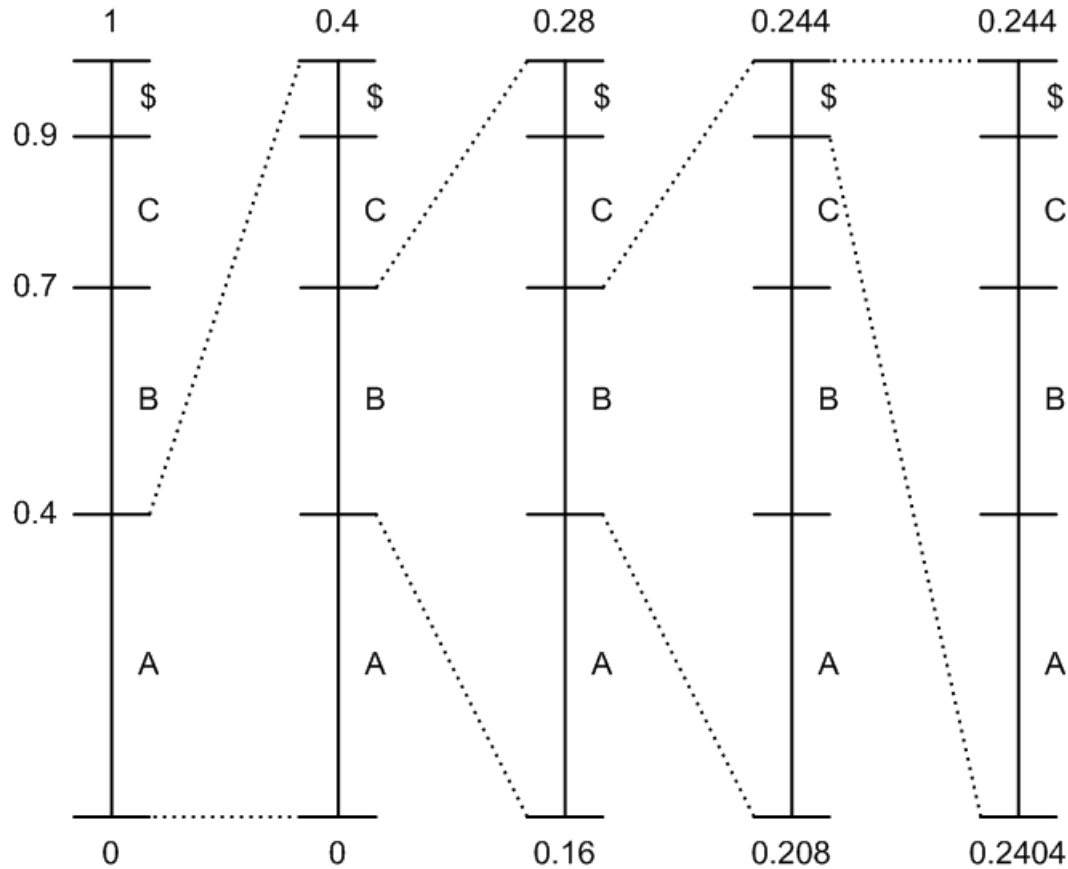
```
low = 0.0;
high = 1.0;
repeat {
    input s;
    range = high - low;
    high = low + range * highrange[s];
    low = low + range * lowrange[s];
} until s = $;
output any number in [low, high);
```

# Αριθμητική (4 από 10)

- Λειτουργία αλγορίθμου
  - Lowrange[]: κάτω όρια πιθανοτήτων
  - Highrange[]: άνω όρια πιθανοτήτων
  - Η είσοδος κωδικοποιείται με διάστημα
    - Αρχικά το διάστημα είναι  $[0,1)$
    - Σε κάθε βήμα το διάστημα περιορίζεται
    - Ανάλογα με το σύμβολο εισόδου
    - Όσο προχωράει η είσοδος, μικραίνει το διάστημα



# Αριθμητική (5 από 10)



- Παράδειγμα αριθμητικής κωδικοποίησης

# Αριθμητική (6 από 10)

- Υπολογισμός εξόδου
  - Θέλουμε έναν αριθμό μέσα στο διάστημα
  - Αλλά με τα λιγότερα δυνατά ψηφία!
  - Αρχίζουμε με 0, και προσθέτουμε ψηφία
  - Επισυνάπτουμε μονάδα στο δεξί άκρο
    - Αν είμαστε πάνω από το άνω όριο, βάζουμε μηδέν
    - Αν είμαστε πάνω από το κάτω όριο, σταματάμε
  - Δεν χρειάζεται να στείλουμε το 0,

# Αριθμητική (7 από 10)

- Αλγόριθμος αποκωδικοποίησης

```
input n;
```

```
repeat {
```

```
    find s so that n is in  
    [lowrange[s], highrange[s]);
```

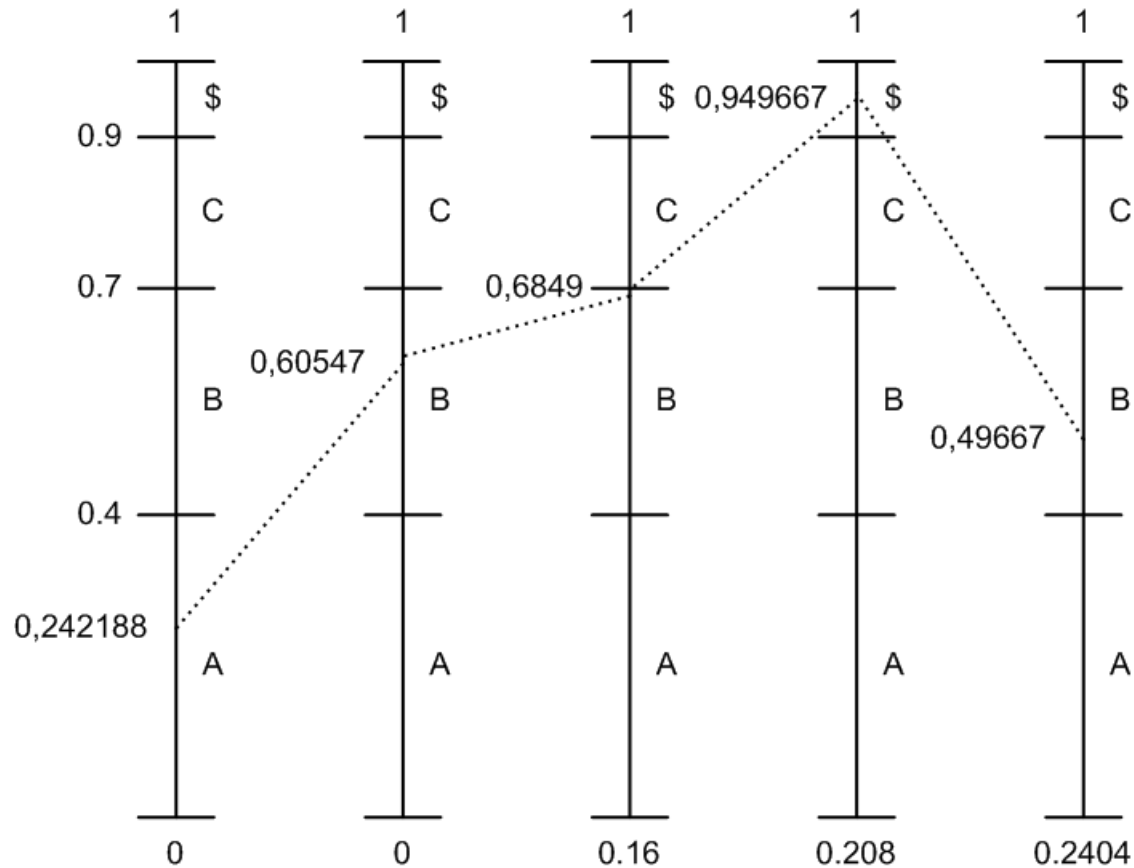
```
    output s;
```

```
    range = highrange[s] - lowrange[s];
```

```
    n = (n - lowrange[s]) / range;
```

```
} until s = $;
```

# Αριθμητική (8 από 10)



- Παράδειγμα αριθμητικής αποκωδικοποίησης

# Αριθμητική (9 από 10)

- Μειονεκτήματα αριθμητικής κωδικοποίησης
  - Διαχείριση αριθμών άγνωστου μήκους
    - Πάρα πολλά δεκαδικά ψηφία
    - Χρήση ειδικών βιβλιοθηκών
- Κωδικοποίηση ομάδων σταθερού μήκους
  - Σπάμε την είσοδο σε μπλοκ
  - Κάθε μπλοκ απαιτεί λιγότερα ψηφία
  - Χάνουμε λίγο σε αποδοτικότητα
  - Δεν χρειάζεται σύμβολο τερματισμού

# Αριθμητική (10 από 10)

- Σε τι χρειάζεται το σύμβολο τερματισμού;
  - Η αποκωδικοποίηση μας δίνει έναν αριθμό
    - Δεν μας δίνει διαστήματα
  - Δεν είναι σαφές πότε αποκωδικοποιήθηκαν όλα
    - Εκτός αν δουλεύουμε με μπλοκ σταθερού μήκους
  - Το τερματικό σύμβολο λύνει το πρόβλημα
    - Αν είμαστε στην περιοχή του, έχουμε τελειώσει
  - Δεν χρειάζεται να είναι πραγματικό σύμβολο
    - Αρκεί να του δώσουμε μία περιοχή πιθανοτήτων

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κωδικοποίηση με παράθυρο

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ

# Γιατί παράθυρο (1 από 2);

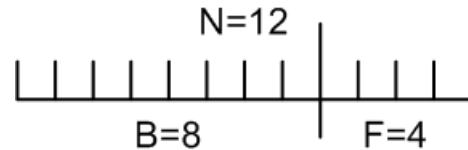
- Περιορισμοί κωδικοποιήσεων εντροπίας
  - VLC: Δουλεύουμε με bits
  - Αριθμητική: Δουλεύουμε με μεγάλους αριθμούς
  - Το όριό τους είναι η εντροπία της πηγής
- Κωδικοποίηση ομάδων συμβόλων
  - Ιδανική, μεταβλητού μήκους
  - Ποιες ομάδες εμφανίζονται συχνά;
  - Πώς μπορούμε να τις κωδικοποιήσουμε;



# Γιατί παράθυρο (2 από 2);

- Σε κάθε στιγμή ο κωδικοποιητής
  - Έχει κωδικοποιήσει την είσοδο που προηγείται
  - Πρέπει να κωδικοποιήσει την είσοδο που έπεται
- Κωδικοποίηση με παράθυρο
  - Ψάχνω να βρω τμήματα της εισόδου...
  - ...που τα έχω δει ήδη κωδικοποιήσει...
  - ...ώστε να αντιστοιχίσω τα νέα με τα παλιά

# Αλγόριθμος LZ77 (1 από 4)



- Αλγόριθμος LZ77 (Lempel-Ziv)
  - Σε κάθε στιγμή βλέπει παράθυρο της εισόδου
    - Αριστερό μέρος: κωδικοποιημένη είσοδο
    - Δεξί μέρος: είσοδος προς κωδικοποίηση
  - Αντικατάσταση προθέματος με τριάδα (O,L,C)
    - O: θέση προθέματος στο αριστερό μέρος
    - L: μήκος του προθέματος που ταιριάζαμε
    - C: πρώτος χαρακτήρας που δεν ταιριάζει

# Αλγόριθμος LZ77 (2 από 4)

|a|c|a|b|b|a|c|a|b|a|a|c|

- Παράδειγμα κωδικοποίησης LZ77
  - Αντικαθιστούμε το baα με (4,2,a)
    - Θέση 4 στο παράθυρο
      - Η πρώτη θέση στο παράθυρο είναι 0
    - Μήκος ταιριάσματος 2
    - Επόμενο σύμβολο a
  - Αν δεν βρούμε ταιρίασμα, θέτουμε μήκος 0

# Αλγόριθμος LZ77 (3 από 4)

|a|c|a|b|b|a|c|a|a|a|c|b|

- Παράδειγμα με επικάλυψη
  - Επέκταση ταιριάσματος στο δεξί μέρος
  - Αντικαθιστούμε το aac με (7,2,c)
- Υλοποίηση κωδικοποίησης LZ77
  - Το παράθυρο συνήθως είναι δύναμη του 2
  - Παράδειγμα: 4096+4096 σύμβολα
  - Δείκτης: 12 bit για όλο το αριστερό μέρος
  - Μήκος: 12 bit για το μήκος του δεξιού μέρους

# Αλγόριθμος LZ77 (4 από 4)

- Εκκίνηση (απο)κωδικοποίησης
  - Υποθέτουμε γνωστό παράθυρο στα αριστερά
- Μειονεκτήματα LZ77
  - Κάθε τριάδα απαιτεί 4-5 byte στο παράδειγμα
  - Το αρχείο μεγαλώνει με κακά ταιριάσματα
  - Οι χαρακτήρες αρχικά κωδικοποιούνται ως (0,0,c)
    - Η κωδικοποίηση ξεκινά με παθητικό!
    - Βελτίωση: αρχικό παράθυρο το σύνολο συμβόλων

# Αλγόριθμος LZSS (1 από 2)

- Αλγόριθμος LZSS (Storer και Szymanski)
  - Αποφεύγει τα μειονεκτήματα του LZ77
  - Διαφέρει στο τι παράγεται στην έξοδο
  - Δύο περιπτώσεις: ταίριασμα ή χαρακτήρας
  - Το πρώτο bit της εξόδου διακρίνει τι είναι
    - Είτε (O,L): θέση O, μήκος L
    - Είτε C: χαρακτήρας C χωρίς ταίριασμα
  - Η τριάδα σπάει σε δύο κομμάτια

# Αλγόριθμος LZSS (2 από 2)

- Υλοποίηση κωδικοποίησης LZSS
  - Δεν θέλουμε κωδικούς των 9 bit!
  - Χωρίζουμε την έξοδο σε ομάδες οκτώ κωδικών
  - Το πρώτο byte περιγράφει τους κωδικούς
    - Ένα bit για κάθε κωδικό
    - Δείχνει αν είναι ταίριασμα ή χαρακτήρας
    - Τα υπόλοιπα byte ερμηνεύονται αντίστοιχα
  - Διαβάζουμε συνέχεια ολόκληρα byte (ή λέξεις)

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**



**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Κωδικοποίηση με λεξικό

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης





# Αλγόριθμος LZ78 (1 από 2)

- Λεξικό αντί για παράθυρο
  - Δυνατότητα ταιριάσματος με πολύ παλιά είσοδο
  - Μέγιστο πρόθεμα εισόδου που υπάρχει στο λεξικό
  - Αντικατάσταση προθέματος με το ζεύγος (P,C)
    - P: θέση του προθέματος στο λεξικό
    - C: πρώτος χαρακτήρας που δεν ταιριάζει
  - Πρόθεμα + χαρακτήρας μπαίνουν στο λεξικό
  - Ο αποκωδικοποιητής χτίζει το ίδιο λεξικό
    - Χρήση του λεξικού για την αποκωδικοποίηση

# Αλγόριθμος LZ78 (2 από 2)

Είσοδος	Έξοδος	Λεξικό
a	(0,a)	Θέση 1: a
aa	(1,a)	Θέση 2: aa
b	(0,b)	Θέση 3: b
ba	(3,a)	Θέση 4: ba
ab	(1,b)	Θέση 5: ab

- Παράδειγμα: είσοδος aaabbaab
  - Το λεξικό σταδιακά αποκτά μεγάλες συμβολοσειρές
    - Στον LZ77 κάνουμε αμέσως μεγάλα ταιριάσματα
  - Ο αποκωδικοποιητής χτίζει το λεξικό από την έξοδο
    - Οι αναφορές γίνονται σε προηγούμενες καταχωρήσεις

# Αλγόριθμος LZW (1 από 8)

- Αλγόριθμος LZW (Welch)
  - LZ78: ακολουθεί τη λογική του LZ77
    - Σε κάθε ταίριασμα προσθέτει επόμενο χαρακτήρα
    - Εγγυημένη πρόοδος ακόμη και χωρίς ταίριασμα
  - Ο LZW παράγει μόνο δείκτες, όχι χαρακτήρες
    - Το λεξικό αρχικοποιείται με όλα τα σύμβολα
    - Οι συμβολοσειρές χτίζονται από αυτά τα σύμβολα
    - Πρέπει όμως κάπως να μεγαλώνει το λεξικό

# Αλγόριθμος LZW (2 από 8)

- Κωδικοποίηση LZW
  - Μέγιστο πρόθεμα εισόδου που είναι στο λεξικό
  - Αντικαθίσταται από το δείκτη στο λεξικό
    - Δεν καταναλώνουμε τον επόμενο χαρακτήρα
  - Προσθέτουμε πρόθεμα + επόμενο χαρακτήρα
    - Σταδιακή αύξηση μήκους συμβολοσειρών στο λεξικό
  - Η είσοδος προχωράει μετά το πρόθεμα
    - Ο επόμενος χαρακτήρας γίνεται αρχή της εισόδου

# Αλγόριθμος LZW (3 από 8)

```
input s;
while not EOF {
    input c;
    if [s+c] is in dictionary
        s = [s+c];
    else {
        output code(s);
        add [s,c] to dictionary with next code;
        s = c; }
}
output code(s);
```

# Αλγόριθμος LZW (4 από 8)

Είσοδος	Έξοδος	Λεξικό
		Θέση 1: a
		Θέση 2: b
a+a	1	Θέση 3: aa
aa+b	3	Θέση 4: aab
b+b	2	Θέση 5: bb
b+a	2	Θέση 6: ba
aab+b	4	Θέση 7: aabb

- Παράδειγμα: είσοδος aaabbaabb
  - Το λεξικό έχει αρχικά όλα τα σύμβολα

# Αλγόριθμος LZW (5 από 8)

- Αποκωδικοποίηση LZW
  - Διαβάζουμε τον κωδικό
  - Αν υπάρχει στο λεξικό, τον αντικαθιστούμε
    - Δεν εισάγουμε τρέχουσα συμβολοσειρά στο λεξικό
    - Δεν γνωρίζουμε τον επόμενο χαρακτήρα!
  - Εισάγουμε προηγούμενη + πρώτο χαρακτήρα
    - Τώρα μόνο γνωρίζουμε ποιος ήταν ο χαρακτήρας
    - Ο αποκωδικοποιητής είναι ένα βήμα πίσω

# Αλγόριθμος LZW (6 από 8)

- Αν ο κωδικός δεν υπάρχει στο λεξικό;
  - Έστω ότι η είσοδος είναι ο τελευταίος κωδικός
  - Δεν τον έχουμε προσθέσει ακόμη στο λεξικό!
    - Δεν γνωρίζουμε ακόμη τον επόμενο χαρακτήρα
  - Η συμβολοσειρά είναι της μορφής C???C
    - Μοναδικός τρόπος να συμβεί το παραπάνω
    - Άρα παίρνουμε την προηγούμενη συμβολοσειρά
    - Και προσθέτουμε τον πρώτο της χαρακτήρα στο τέλος



# Αλγόριθμος LZW (7 από 8)

```
s = NIL; // previous string
while not EOF {
    input c;
    entry = string(c); // current string
    if entry not in dictionary
        entry = s + s[0];
    output entry;
    if (s != NIL) // only happens once
        add [s,entry[0]] to dictionary with next code;
    s = entry; // current string becomes previous
}
```

# Αλγόριθμος LZW (8 από 8)

Είσοδος	Έξοδος	Λεξικό
		Θέση 1: a
		Θέση 2: b
1	a	
3	aa	Θέση 3: aa
2	b	Θέση 4: aab
2	b	Θέση 5: bb
4	aab	Θέση 6: ba

- Παράδειγμα αποκωδικοποίησης LZW
  - Ο κωδικός 3 δείχνει σε κενή θέση
  - Πρέπει να είναι παλιά + πρώτος χαρακτήρας (a+a)

# Βελτιστοποίηση (1 από 3)

- Υλοποίηση λεξικού LZ78/LZW
  - Το λεξικό μεγαλώνει σε κάθε βήμα
  - Επεκτεινόμενοι δείκτες
    - Ξεκινάμε με δείκτες έστω 4 bit (16 θέσεις)
    - Όποτε γεμίζει το λεξικό, προσθέτουμε 1 bit
  - Τι γίνεται όταν το λεξικό μεγαλώσει πολύ;
    - Είτε σταματάμε να δεχόμαστε συμβολοσειρές
    - Είτε πετάμε τις λιγότερο χρησιμοποιημένες

# Βελτιστοποίηση (2 από 3)

- Συμπύεση λεξικού LZ78/LZW
  - Κάθε νέα είσοδος επεκτείνει μία παλιά
    - Κατά έναν χαρακτήρα ακριβώς
  - Αποθήκευση δείκτη σε παλιά συμβολοσειρα
    - Και επιπλέον του νέου χαρακτήρα
  - Πώς μπορεί να γίνει αυτό αποδοτικά;
    - Στην κωδικοποίηση διατρέχω το λεξικό
    - Ψάχνω το μεγαλύτερο δυνατό ταίριασμα

# Βελτιστοποίηση (3 από 3)

- Δένδρο trie
  - Κάθε κόμβος έχει χαρακτήρες ως παιδιά
  - Κάθε ταίριασμα είναι ένα μονοπάτι
    - Κάθε σύμβολο με πάει ένα επίπεδο παρακάτω
    - Όταν φτάσω σε άκρο, βρήκα μέγιστο ταίριασμα
    - Ο επόμενος χαρακτήρας προστίθεται ως φύλλο
  - Επιτάχυνση της αναζήτησης
    - Η αποκωδικοποίηση κάνει τη δομή πιο περίπλοκη

**ΟΙΚΟΝΟΜΙΚΟ  
ΠΑΝΕΠΙΣΤΗΜΙΟ  
ΑΘΗΝΩΝ**

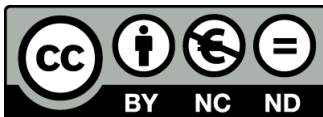


**ATHENS UNIVERSITY  
OF ECONOMICS  
AND BUSINESS**

# Τέλος Ενότητας #9

**Μάθημα:** Τεχνολογία Πολυμέσων, **Ενότητα # 9:** Κωδικοποίηση εντροπίας

**Διδάσκων:** Γιώργος Ξυλωμένος, **Τμήμα:** Πληροφορικής



Ευρωπαϊκή Ένωση  
Ευρωπαϊκό Κοινωνικό Ταμείο



ΥΠΟΥΡΓΕΙΟ ΠΑΙΔΕΙΑΣ & ΘΡΗΣΚΕΥΜΑΤΩΝ, ΠΟΛΙΤΙΣΜΟΥ & ΑΘΛΗΤΙΣΜΟΥ  
ΕΙΔΙΚΗ ΥΠΗΡΕΣΙΑ ΔΙΑΧΕΙΡΙΣΗΣ

Με τη συγχρηματοδότηση της Ελλάδας και της Ευρωπαϊκής Ένωσης



ΕΥΡΩΠΑΪΚΟ ΚΟΙΝΩΝΙΚΟ ΤΑΜΕΙΟ