Chương 4: Mã hóa nguồn

4.1. Cơ bản về mã hóa

- Tại một thời điểm, nguồn tạo ra một ký hiệu từ bảng ký hiệu của nguồn
- Thông thường, bảng chữ là hữu hạn
  - S = {s1, s2, …, sq} Ở đây q là ||S|| hoặc sô ký hiệu của nguồn S
- Mã hóa: Sử dụng một tập hữu hạn các ký hiệu mã để biểu diễn các ký hiệu của nguồn
  - Có thể biểu diễn tập ký hiệu mã bởi tập X = {x1, x2,…,xr}. Ví dụ với mã BCD, X = (0,1), r=2
  - r là ||X|| hay số ký hiệu mã khác nhau
    - được gọi là cơ số của mã
    - r = 2 : mã nhị phân
    - r ≠ 2: mã r trị

4.1. Cơ bản về mã hóa

- Thông thường số ký hiệu nguồn của tập nguồn lớn hơn số ký hiệu mã của tập mã q>>r
  - Cần phải tạo tổ hợp các ký hiệu mã để biểu diễn một ký hiệu của nguồn hay mã hóa một tin của nguồn
    - Sử dụng luật tạo tổ hợp hay còn gọi luật tạo từ mã (luật tạo từ)
    - Tổ hợp có thể là tổ hợp thỏa mãn luật này. Chỉ tổ hợp có thể mới được dùng để mã hóa
    - Ví dụ với mã BCD, luật tạo tổ hợp Mỗi tổ hợp là một chuỗi dài 4 ký hiệu nhị phân
  - Mỗi tổ hợp có thể được dùng để biểu diễn (mã hóa) một ký hiệu nguồn
    - Mỗi tổ hợp có thể sẽ được gán cho một tin và tổ hợp có thể có chứ tin này sẽ được gọi là từ mã (mã, từ)
    - Tổ hợp có thể không được dùng để biểu diễn một tin nào được gọi là tổ hợp thừa hay tổ hợp cấm
    - Ví dụ, mã BCD mã số 0 bằng tổ hợp 0000, số 1 bằng 0001.., số 9 bằng 1001 và có 6 tổ hợp thừa 1010,.., 1111
- Mã không có tổ hợp thừa được gọi mã đầy

4.1. Cơ bản về mã hóa

- Luật mã hóa là luật gán 1 tin vào 1 tổ hợp có thể để tạo ra từ mã hay luật ánh xạ 1 tin vào 1 từ mã si → C(si)
  - C(si) là từ mã của tin si Hay C(si) là tổ hợp có thể chứa tin si.
  - C(si) = xi1..xil. ở đây, l là số ký hiệu mã có trong từ mã
- Luật mã hóa thường được biểu diễn bởi bảng mã là bảng mô tả quan hệ si → C(si)
- Độ dài từ mã là số ký hiệu mã có trong từ mã và được ký hiệu là l
  - Nếu độ dài từ mã là giống nhau (cùng một l) với mọi từ mã thì mã được gọi là mã đều hay mã có độ dài cố định
  - Nếu mỗi từ mã có độ dài khác nhau thì mã được gọi là mã có độ dài thây đổi hay không đều
  - Ví dụ, BCD độ dài các từ mã đều là 4 nên nó là mã đều
- Bộ mã hay mã hiệu là tập các từ mã của tất cả các tin của nguồn

4.1. Cơ bản về mã hóa (Cont.)

- Quá trình mã hóa:
  - Lần lượt thay mỗi ký hiệu nguồn của bản tin bằng một từ mã
  - Sau quá trình mã hóa bản tin được chuyển thành chuỗi các ký hiệu mã, thường được gọi là bản mã
  - Ví dụ, sử dụng mã BCD, bản tin 23 được chuyển thành 00100011
- Quá trình giải mã:
  - Tách chuỗi mã nhận được thành các từ mã - quá trình tách từ mã hay phân tách mã
  - Chuyển mỗi từ mã thành một ký hiệu nguồn - quá trình giải mã
  - Ví dụ: chuỗi ký hiệu mã nhận được 00100011
    - Phân tách mã thành 0010 – 0011
    - Giải mã thành 2-3

4.1. Cơ bản về mã (Cont.)

- Các loại mã:
  - Mã duy nhất (Singular code): Mỗi từ mã là duy nhất cho một tin hay mỗi từ mã chỉ xuất hiện duy nhất một lần trong bảng mã
    - Ví dụ: 0001 đã là từ mã của số "1" thì 0001 không được dùng làm từ mã cho ký hiệu nguồn khác
    - Thông thường các mã trong các hệ thống truyền thông là duy nhất. (trừ trong ngôn ngữ tự nhiên có tồn tại các từ đa ngĩa)
  - Mã giải mã được hay giải mã duy nhất :
    - Là mã duy nhất
    - Kết quả giải mã trả lại chính bản tin được truyền
      - Từ chuỗi ký hiệu mã nhận được chỉ tồn tại duy nhất một cách tách nó thành chuỗi các từ mã - phân tách được
        - Mã giải mã được: không tồn tại chuỗi ký hiệu mã của chuỗi các từ mã khác nhau lại trùng nhau và mỗi từ mã chỉ tương ứng duy nhất với một tin

4.1. Cơ bản về mã(Cont.)

- Các loại mã:
  - Ví dụ về mã giải mã được:
  - Giả sử có nguồn, S = {s1,s2,s3,s4}, và tập ký hiệu mã, X= {0,1}.
  - Sâu đây là 3 mã.

Source	Code A	Code B	Code C
s_1	0	0	00
s_2	11	11	01
s_3	00	00	10
s_4	11	010	11

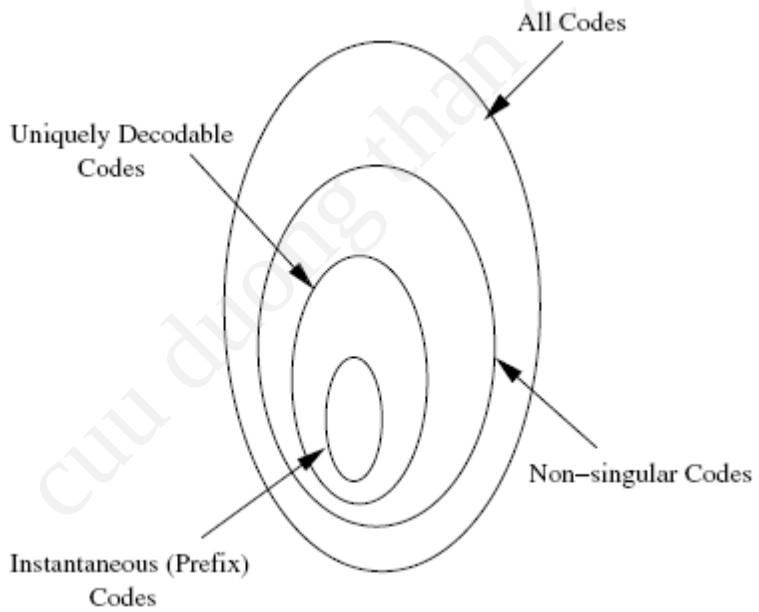
  - Mã A: vi phạm tính duy nhất: “11” từ mã “11” biểu diễn cả 2 tin s2 và s4
  - Mã B: Vi phạm tính phân tách được: Bản tin “s1s3” được mã hóa bởi “000”. Tuy nhiên “000” có thể phân tách thành : “0-0-0”, “0-00”, “00-0”. Bản tin sau giải mã có thể là: “s1-s1-s1”, “s1-s3”, and “s3-s1”
  - Mã C: Giải mã được

4.1. Cơ bản về mã

- Các loại mã:
  - Mã giải mã tức thì:
    - Là mã giải mã được
    - Sau khi nhận được ký hiệu cuối cùng của từ mã, từ mã sẽ được tách ra và việc tách từ mã này là cách tách duy nhất.
    - Mã giải mã tức thì cho phép tách từ mã nhanh nhất nên luôn được dùng trong truyền thông
    - Để có thể giải mã tức thì, mã phải có tính prefix (tính phần đầu, tính tiền tố)
      - Tính prefix thể hiện ở chỗ là không có từ mã nào trùng với prefix của từ mã khác trong bộ mã.
      - Preffix của từ mã là chuỗi ký hiệu mã tính từ ký hiệu đầu (ký hiệu xuất hiện sớm nhất) của từ mã
      - Ví dụ: từ mã 10110 có các preffic 1, 10, 101, 1011, 10110
      - Ví dụ: bộ mã có tính preffix cho nguồn {s1, s2, s3, s4} là {0, 10, 110, 1110}

4.1. Cơ bản về mã (Cont.)

- Các loại mã:



4.1. Cơ bản về mã (Cont.)

- Xây dựng mã có tính prefix:
  - Nguồn có q ký hiêu : cần có q từ mã có độ dài {l1,l2,…,lq}.
  - Thiết kế mã : Độ dài từ mã sẽ được chọn tăng dần
  - Từ mã sẽ là chuỗi ký hiệu có độ dài đã chọn và các từ mã đã được chọn không là prefix của từ mã đang chọn
  - Thực hiện theo cách trên cho đến khi tìm đủ các từ mã cho các tin của nguồn.

4.1. Cơ bản về mã (Cont.)

- Xây dựng mã prefix:

Ví dụ 1. Cần xây dựng bộ mã nhị phân prefix có các từ mã có độ dài 3,2,3,2,2

- Độ dài được sắp xếp lại theo thứ tự tăng dần 2, 2, 2, 3, 3.
- Cho 3 từ mã đầu có độ dài 2:

  00
  01
  10
- Cho hai từ mã có độ dài 3, phần đầu của nó sẽ là 11, và sẽ là 110 và 111.
- Bộ mã đầy đủ sẽ là:

  00
  01
  10
  110
  111

4.1. Cơ bản về mã (Cont.)

- Xây dựng mã prefix:
  - Ví dụ 2. Cần xây dựng bộ mã tam phân prefix có các độ dài 2,3,1,1,2
    - Sắp xếp lại độ dài tăng dần là 1, 1, 2, 2, 3
    - Hai từ mã đầu độ dài 1 là
      0
      1
    - Hai từ mã tiếp sau có độ dài 2 sẽ có phần đầu là 2 và sẽ là:
      20
      21
    - Từ mã tiếp có độ dài 3 sẽ có phần đầu là 22 và từ mã là 220. Vậy bộ mã là
      0
      1
      20
      21
      220

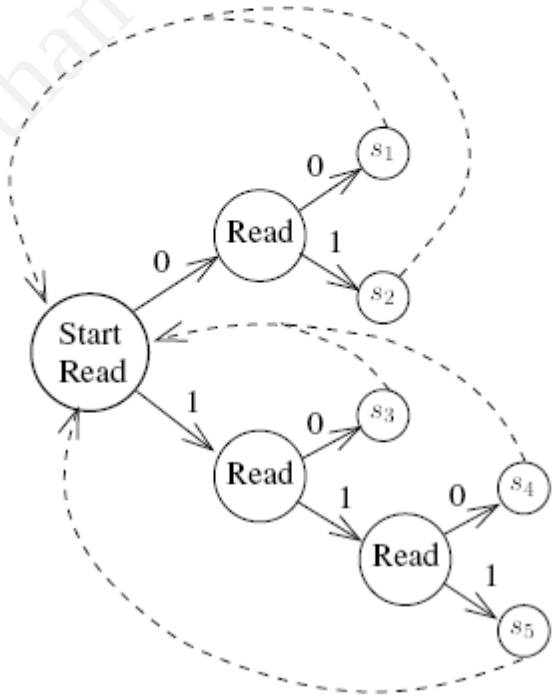
4.1. Cơ bản về mã (Cont.)

- Giải mã prefix:
  - Bắt đầu từ ký hiệu mã đầu tiên nhận được:
    - Tìm từ mã
      - Nếu là ký hiệu cuối của từ mã thì tách từ mã ra rồi tiếp tục nhận thêm ký hiệu mã và lại quay về tìm từ mã
      - Nếu không phải ký hiệu cuối của từ mã thì nhận tiếp ký hiệu mã và quay về tìm từ mã
  - Thuật toán giải mã prefix dùng cho mã có tính prefix là mã thường dùng trong truyền thông.
  - Sử dụng bảng mã để chuyển từ mã tách ra được về tin của nguồn

4.1. Cơ bản về mã (Cont.)

- Giải mã prefix:

Source	Code
s_1	00
s_2	01
s_3	10
s_4	110
s_5	111



4.1. Cơ bản về mã (Cont.)

- Sự nhạy với lỗi ký hiệu của giải mã prefix:
  - Kênh luôn có nhiêu: ký hiệu được truyền sẽ có thể bị sai (lỗi ký hiệu)
  - Trong trường hợp mã có độ dài cố định: để tách từ mã, chỉ cần dựa vào độ dài từ mã (đếm số ký hiệu mã nhận được, nếu bằng độ dài từ mã thì tách từ mã ra). Các từ mã được tach ra không lỗi
  - Trường hợp mã có độ dài thay đổi: Nếu ký hiệu mã thay đổi, từ mã có thể chuyển thành prefix của từ mã khác và sẽ sảy ra lỗi tách từ mã dẫn đến sai giải mã
  - Mã có tính prefix có độ dài thay đổi không được dùng cho kênh truyền có lỗi
    - Để giải quyết tình trạng trên, một số ký hiệu đặc biệt được thêm vào cuối mỗi từ mã:
      - Chuỗi ký hiệu đặc biệt này không được quyền là chuỗi ký hiệu mã bất kỳ trong từ mã
      - Chuỗi ký hiệu đặc biệt này khó bị lỗi chuyển thành chuỗi ký hiệu mã trong từ mã
      - Chuỗi ký hiệu mã này sẽ được gọi là đấu phân tách
      - Mã có dấu phân tách ở cuối mỗi từ mã được gọi là mã có dấu phân tách. Việc tách từ mã sẽ thực hiện thông qua tìm dấu phân tách của từ mã.
  - Chú ý:
    - Kênh không nhiễu dùng mã có tính prefix
    - Kênh có nhiễu, dùng mã đều hoặc mã có dấu phân tá

4.1. Cơ bản về mã (Cont.)

- Bất đẳng thức Kraft:
  - Cung cấp giới hạn về độ dài từ mã khi thiết kế mã có tính prefix cũng như kiểm tra bộ mã có thỏa mãn tính prefix không.
  - Điều kiện cần và đủ cho bộ mã có cơ số r và có q từ mã dài l1,l2…,lq là mã giải mã tức thì là thỏa mãn bất đẳng thức Kraft:

$$\sum_{i=1}^{q} r^{-l_i} \leq 1$$

4.1. Cơ bản về mã (Cont.)

- Bất đẳng thức Kraft:
  - Ví dụ : Giả sử có các bộ mã nhị phân (r = 2):

Source	Code A	Code B	Code C
s_1	0	0	0
s_2	100	100	10
s_3	110	110	110
s_4	111	11	11

  - Mã nào thỏa mãn bất đẳng thức Kraft?

4.1. Cơ bản về mã (Cont.)

- Bất đẳng thức Kraft:
  - Ví dụ:

Source	Code A	Code B	Code C
s_1	0	0	0
s_2	100	100	10
s_3	110	110	110
s_4	111	11	11

- Mã A thỏa mãn vì:

$$\sum_{i=1}^{4} 2^{-l_i} = 2^{-1} + 2^{-3} + 2^{-3} + 2^{-3} = \frac{7}{8} \leq 1$$

- Mã B thỏa mãn nhưng không prefix
  - từ mã của s4 là prefix của s3
  - Có thể sửa từ mã của s4, ví dụ: 0, 110, 111, 10

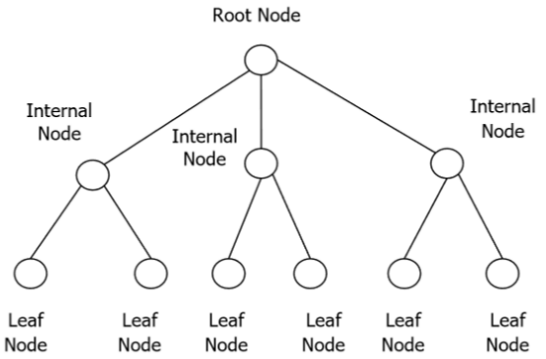
$$\sum_{i=1}^{4} 2^{-l_i} = 2^{-1} + 2^{-3} + 2^{-3} + 2^{-2} = 1 \leq 1$$

Mã C không thỏa mãn vì

$$\sum_{i=1}^{4} 2^{-l_i} = 2^{-1} + 2^{-2} + 2^{-3} + 2^{-2} = \frac{9}{8} > 1$$

4.1. Cơ bản về mã(Cont.)

- Cây mã:
  - Đồ thị hình cây biểu diễn bộ mã
    - Bắt đầu từ nút gốc (Cũng được gọi nút mức 0)
    - Mỗi nút tỏa ra nhiều nhất r nhánh (r là cơ số của mã)
    - Mỗi nhánh kết thúc ở một nút lá
  - Mỗi từ mã được biểu diễn bằng 1 đường từ nút gốc
    - Mỗi nhánh biểu diễn 1 ký hiệu mã. Ký hiệu đầu của từ mã ứng với nhánh tỏa ra từ nút gốc. Các nhánh tiếp theo của đường tương ứng với các ký hiệu mã tiếp theo của từ mã
    - Nút kết thúc của nhánh biểu diễn ký hiệu mã cuỗi cùng của từ mã là nút kết thúc của từ mã (nút cuối). Có thể coi nút cuối này đại diện cho từ mã (mã hóa 1 tin của nguồn)
  - Hệ quả:
    - Mã có tính prefix: Nút cuỗi là nút lá
    - Mã có độ dài cố định: Các nút cuối là nút lá và ở cùng mức
    - Mã đầy có tính prefix: Mỗi nút tỏa ra đủ r nhánh



4.2. Độ dài trung bình của từ mã và mã có độ dài trung bình ngắn nhất (compact code)

- Gọi {pi : i=1,..,q} là xác suất xuất hiện của các ký hiệu của nguồn có q ký hiệu.
- Gọi {li : i=1,..,q} là độ dài các từ mã mã hóa các tin của nguồn
- Độ dài trung bình của từ mã, L, được tính theo công thức:

$$L = \sum_{i=1}^{q} P_i l_i$$

- Vì từ mã dài li mã hóa tin si có xác suất pi của nguồn nên độ dài li có xác suất xuất hiện cũng là pi

4.2. Độ dài trung bình của từ mã và mã có độ dài trung bình ngắn nhất

- Mã có độ dài ngắn nhất là mã giải mã được và có độ dài trung bình không dài hơn độ dài trung bình của các mã giải mã được khác cho cùng 1 nguồn
- Ví dụ:

Source	P_i	Code A	Code B
s_1	0.5	00	1
s_2	0.1	01	000
s_3	0.2	10	001
s_4	0.2	11	01

- Mã A có độ dài trung bình băng 2 kh (ký hiệu mã)/ tin
- Mã B có độ dài trung bình: 0.5x1 + 0.1x3 + 0.2x3 + 0.2x2 = 1.8 kh/tin
- Mã B có độ dài trung bình ngắn hơn

4.3. Giới hạn dưới của độ dài trung bình từ mã

- Mã giải mã tức thì r trị của nguồn S ={s1,..,sq}, có độ dài trung bình của từ mã L. Độ dài trung bình của từ mã bé nhất bằng Entropy của nguồn Hr(S) (Hr(S) là Entropy của nguồn với cơ số của hàm logarithm là r). L>= Hr(S)> Hr(s) là Entropy của nguồn có cơ số của hàm logarithm tính theo cơ số mã r.
  - Khi mã hóa, lượng tin của tin si có xác suất p(si) là - logr(p(si)) phải được chuyển sang li ký hiệu mã mã hóa tin si.
  - Để số ký hiệu mã li là tối thiểu thì lượng tin có thể chứa trong mỗi ký hiệu mã phải đạt cực đại (nguồn các ký hiệu mã phải đẳng xác xuất) và bằng logr(r) = 1 đơn vị thông tin tính theo cơ số của hàm logarithm bằng cơ số mã r.
  - Đê mã hóa không mất mát thông tin li >= -logr(p(si)) → lấy trung bình hai vế sẽ có L>= Hr(S)
- Hiệu suất của mã được định nghĩa:

$$\eta = \frac{H_r(S)}{L} \times 100\%$$

- Mã có L = Hr(S) hay hiệu suất mã đạt 100% được gọi là mã tối ưu

4.3. Giới hạn dưới của độ dài trung bình từ mã (Cont.)

- η Code efficiency:
  - Special source:
    - Example: 4-symbol source
      - Symbol probabilities are of the form $P_i = (\frac{1}{2})^{l_i}$
      - $l_1 = 3$, $l_2 = 2$, $l_3 = 1$ and $l_4 = 3$
      - A 100% efficient compact binary code can be designed because L = H(S) = 1.75
      - A code can be

Source A	P_i
s_1	0.125
s_2	0.25
s_3	0.5
s_4	0.125

Source A	Code A
s_1	110
s_2	10
s_3	0
s_4	111

4.4. Định lý mã hóa cho kênh không nhiễu của Shannon (định lý mã hóa nguồn)

- Kênh không nhiễu có thể coi là kênh có tốc độ truyền (số đơn vị thông tin truyền không bị sai/ đơn vị thời gian) chỉ phụ thuộc vào tốc độ lập tin của nguồn đặt vào đầu vào của nó. Tốc độ truyền lớn nhất của kênh hay thông lượng của kênh không nhiễu sẽ bằng tốc độ lập tin lớn nhất của nguồn đặt vào đầu vào kênh.
- Khi sử dụng mã hóa mỗi tin của nguồn ban đầu chuyển thành một từ mã có độ dài trung bình L >= Hr(S) theo giới hạn dưới của độ dài trung bình của từ mã. Shannon đưa ra định lý mã hóa cho kênh không nhiễu với nôi dung: Hr(S) <= L <= Hr(S) +1.
- Định lý này còn được gọi là giới hạn về độ dài trung bình của từ mã của mã tối ưu.
- Độ dài trung bình của từ mã sẽ là tối thiểu và mã là mã tối ưu chỉ với một số nguồn đặc biệt đảm bảo L = Hr(S), ví dụ nguồn có phân bố xác suất sao cho độ dài từ mã đúng bằng lượng tin chứa trong tin, li = - logr(p(si)).

4.4. Định lý mã hóa cho kênh không nhiễu của Shannon

- Nếu nguồn không phải nguồn đặc biệt nêu trên, theo Shannon có thể đạt được hiệu suất mã 100% hay tạo được mã tối ưu bằng cách mở rộng nguồn.
  - Mở rộng nguồn n lần:
    - Nguồn mở rộng n lần của nguồn S, Sn, có mỗi tin là một chuỗi có n tin của nguồn S.
    - Entropy của nguồn mở rộng, Hr(Sn) = nHr(S)
    - Độ dài trung bình của từ mã mã hóa mỗi tin của nguồn mở rộng Ln = nL
  - Khi mở rộng nguồn n lần: Để có mã có độ dài trung bình ngắn nhất, mong muốn
    - Hr(S) <= Ln <= Hr(Sn) +1 => nHr(S) <= nL <= nHr(S) + 1
    - Hr(S) <= L <= Hr(S) + 1/n.
      - Khi n tiến tới vô cùng thì mã cho nguồn mở rộng sẽ là mã tối ưu hay hiệu suất mã đạt 100%
      - Hr(S) + 1 chính là giới hạn trên của độ dài trung bình củan)) từ mã của nguồn S .

4.4. Định lý mã hóa cho kênh không nhiễu của Shannon (Cont.)

- Example:
  - binary coding scheme for a binary source

Source	P_i	Compact Code
s_1	0.8	0
s_2	0.2	1

- H(S) = -0.8 $log_2$0.8 – 0.2 $log_2$0.2 = 0.772 bits/symbol
- L = 1 bit/ symbol
- Efficiency = 77.2%

4.4. Định lý mã hóa cho kênh không nhiễu của Shannon (Cont.)

Source	P_i	Compact Code
s_1	0.8	0
s_2	0.2	1

Source	P_i	Compact Code
$s_1 s_1$	0.64	0
$s_1 s_2$	0.16	10
$s_2 s_1$	0.16	110
$s_2 s_2$	0.04	111

- Example:
  - binary coding scheme for a binary source
    - H(S) = -0.8 $log_2$0.8 – 0.2 $log_2$0.2 = 0.772 bits/symbol
    - L = 1 bit/ symbol
    - Efficiency = 77.2%
  - second extension:
    - Each symbol of second extension source is sisj (si, sj ϵ S)
    - L_2 = (0.64)1 + (0.16)2 + (0.16)3 + (0.04)3 = 1.56
    - L_2/n = L_2/2 = 0.78
    - H(S)= -0.64 $log_2$0.64 - 0.16 $log_2$0.16 - 0.16 $log_2$0.16 - 0.04 $log_2$0.04 ≈ 0.722
    - Efficiency = 0.722/0.78 ≈ 92.6%

4.5. Mã tối ưu và thông lượng của kênh

- Mã nguồn sẽ được dùng cho kênh không nhiễu.
  - Thông lượng bằng lượng tin tương hỗ cực đại. Với kênh không nhiễu, lượng tin tương hỗ cực đại bằng Entropy cực đại của nguồn được đặt vào đầu vào kênh.
  - Mã nguồn chuyển nguồn S thành mã X có độ dài trung bình các từ mã là L tối thiểu và lượng tin chứa trong mỗi ký hiệu mã đạt cực đại hay H(X) đạt cực đại. Thông lượng của kênh truyền mã C = H(X) = 1 đơn vị thông tin tính theo cơ số hàm logarthm là cơ số ma r và kênh lượng của kênh truyền các tin của nguồn ban đầu bằng L (coi nhịp tạo tin là đơn vị)
  - Mã hóa nguồn cho độ dài trung bình của từ mã là tối thiểu nên số ký hiệu mã sử dụng để mã hóa hay biểu diễn một bản tin sẽ là tối thiểu → Mã hóa nguồn còn được gọi là nén dữ liệu, nén số lượng phần tử dữ liệu để biểu diễn thông tin của một bản tin.

4.6. Mã hóa nguồn với mã có độ dài thay đổi

- Mã hóa nguồn:
  - Sử dụng số ký hiệu mã tối thiểu để biểu diễn mỗi tin của nguồn để độ dài trung bình từ mã hỏa mãn Hr(S) <= L <= Hr(S) + 1
  - Mã phải là mã có tính prefix
  - Độ dài từ mã tỷ lệ nghich với xác suất xuất hiện của tin được mã hóa (Từ mã là vật chứa thông tin của tin có lượng tin -log(p(xi)))
- Ba điều kiện trên chính là 3 yêu cầu đặt ra cho mỗi thuật toán tìm bộ mã nguồn cho một nguồn S = {si: i=1,..,q} có phân bố xác suất P(S) = {p(si)}.
- Một mã nguồn rất phổ biến có độ dài từ mã thay đổi là mã Huffman

4.6.1 Mã Huffman

- Mã Huffman:
  - Đề xuất bởi Huffman
  - Gán cho mỗi tin một từ mã có độ dài tỷ lệ nghịch với xác suất xuất hiện của tin
  - Mã tạo ra là mã thỏa mãn giới hạn về độ dài trung bình và có tính prefix
  - Thuật toán tìm bộ mã: Liên tiếp thực hiện rút gọn nguồn từ q ký hiệu nguồn về còn r ký hiệu

4.6.1. Mã Huffman

- Thuật toán rút gọn nguồn:
  - Giả sử nguồn S có q tin, (si: i=1,..,q) có xác suất xuất hiện mỗi tin (p(si): i=1,..,q)
  - Xếp các tin theo thứ tự xác suất các tin giảm dần và giả sử sau sắp xếp có p(s1)>=p(s2)>=..>=p(sq)
  - Thay r tin cuối trong dãy các tin được sắp xếp bằng 1 tin mới s' có xác suất xuất hiện là tổng các tin được nhóm vào
  - Nguồn nhận được sau rút gọn được ký hiệu S1.
  - Tiếp tục rút gọn nguồn thành S2, S3,..bằng thuất toán sắp xếp và nhóm tương tự cho đến khi nguồn mới chỉ còn r tin

4.6.1. Mã Huffman

- Thuật toán rút gọn nguồn chỉ có thể rút gọn một nguồn về nguồn mới có chính xác r tin, nếu nguồn ban đầu có số tin q = r + a(r-1) tin. Ở đây a là số nguyên không âm.
  - Với r = 2, mã nhị phân, điều kiện trên đúng với mọi q >= 0
  - Với r > 2, nếu a = (q-r)/(r-1) không nguyên (không thỏa mãn điều kiện trên) thì cần phải thêm một số tin ‘ảo’ có xác suất bằng 0 vào nguồn trước khi rút gọn nguồn.
  - Số tin cần thêm vào phải đảm bảo số tin của nguồn được bổ sung các tin này là q’ = r + a’(r-1), a’ là số nguyên. Vậy a’ phải là số nguyên bé nhất nhỏ hơn hoặc bằng a. (Số tin được thêm vào là q’ - q).

4.6.1 Mã Huffman

- Xác định từ mã Huffman cho mỗi tin của nguồn:
  - Tại mỗi bước rút gọn nguồn, mỗi tin trong r tin cuối (được thay bằng một tin mới s') sẽ được gán tùy ý vào 1 ký hiệu mã (mã cơ số r sẽ có r ký hiệu mã).
  - Từ mã của mỗi tin của nguồn sẽ là chuỗi các ký hiệu mã gán cho chính tin này của nguồn và các tin mới (s') chứa nó. Thứ tự các ký hiệu mã trong từ mã ngược với thứ tự thay thế tạo tin mới.

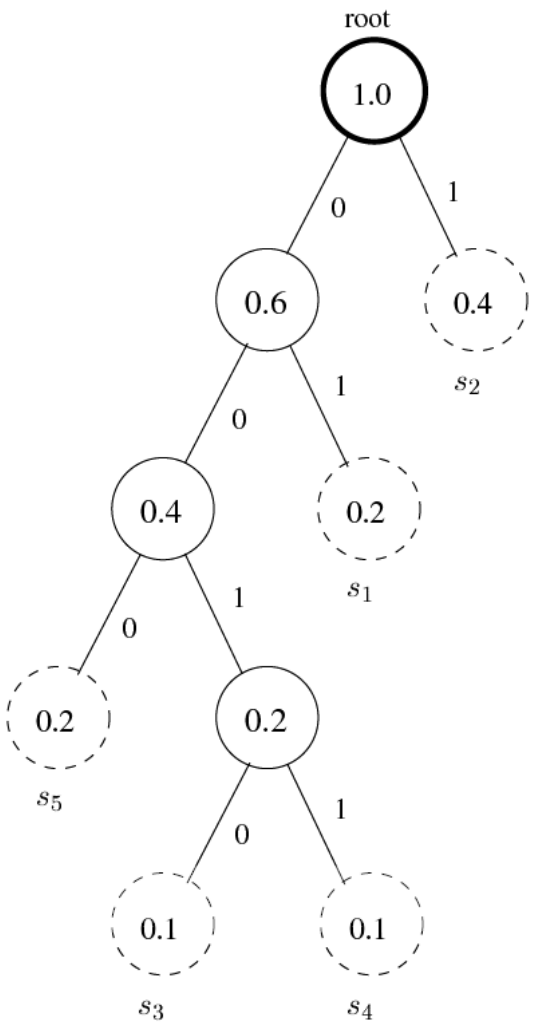
4.6.1. Mã Huffman (Cont.)

- Ví dụ:
  - Nguồn có:

$$P(s_1) = 0.2 \quad P(s_2) = 0.4 \quad P(s_3) = 0.1 \quad P(s_4) = 0.1 \quad P(s_5) = 0.2$$

  - Áp dụng thuật toán tìm mã Huffman:

	S	S_1	S_2	S_3
s_2	0.4 *1*	0.4 *1*	0.4 *1*	0.6 *0*
s_1	0.2 *01*	0.2 *01*	0.4 *00*	0.4 *1*
s_5	0.2 *000*	0.2 *000*	0.2 *01*	
s_3	0.1 *0010*	0.2 *001*		
s_4	0.1 *0011*			



4.6.1. Mã Huffman (Cont.)

- Example:
  - Source has q=11 :

$$P(s_1) = 0.16 \;\; P(s_2) = 0.14 \;\; P(s_3) = 0.13 \;\; P(s_4) = 0.12 \;\; P(s_5) = 0.10$$

$$P(s_6) = 0.10 \;\; P(s_7) = P(s_8) = 0.06 \;\; P(s_9) = 0.05 \;\; P(s_{10}) = P(s_{11}) = 0.04$$

  - Apply Huffman algorithm with r=4:
    - $\alpha = \frac{q-r}{r-1}$ =2.33 (not integer value) → $\lceil \alpha \rceil$ = 3
    - Get new q = 13 according to $q = r + \lceil \alpha \rceil (r-1)$
      - Add new dummy s12,s13 with P(s12) = P(s13) =0

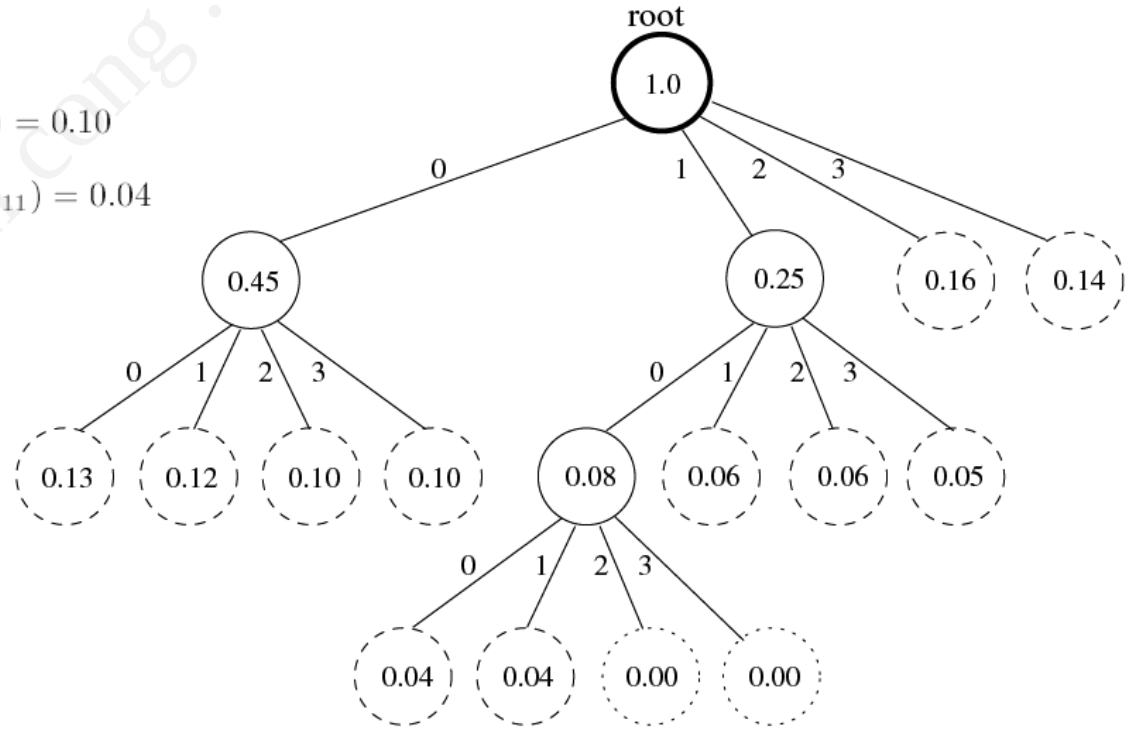
4.6.1. Mã Huffman (Cont.)

- Example:
  - Original Source :

$$P(s_1) = 0.16 \;\; P(s_2) = 0.14 \;\; P(s_3) = 0.13 \;\; P(s_4) = 0.12 \;\; P(s_5) = 0.10$$

$$P(s_6) = 0.10 \;\; P(s_7) = P(s_8) = 0.06 \;\; P(s_9) = 0.05 \;\; P(s_{10}) = P(s_{11}) = 0.04$$

S		S_1		S_2		S_3	
0.16	2	0.16	2	0.25	1	0.45	0
0.14	3	0.14	3	0.16	2	0.25	1
0.13	00	0.13	00	0.14	3	0.16	2
0.12	01	0.12	01	0.13	00	0.14	3
0.10	02	0.10	02	0.12	01		
0.10	03	0.10	03	0.10	02		
0.06	11	0.08	10	0.10	03		
0.06	12	0.06	11				
0.05	13	0.06	12				
0.04	100	0.05	13				
0.04	101						
0.00	102						
0.00	103						



4.6.2. Mã số học - Thuật toán mã hóa

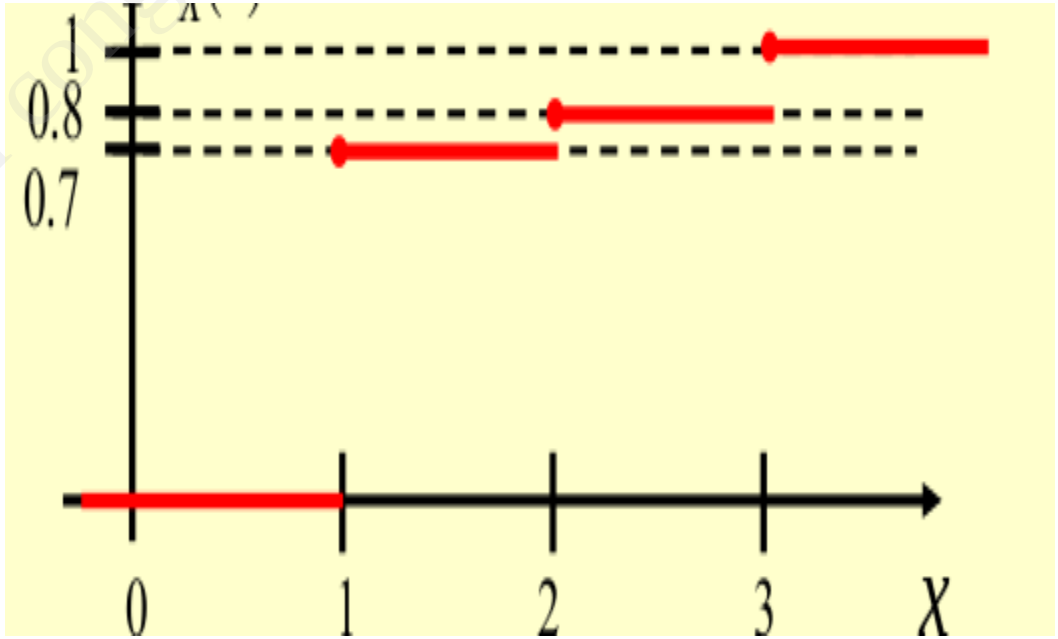
- Bản tin rỗng sẽ được coi tương ứng với một điểm bất kỳ trong đoạn [0,1). Điểm đầu I(0) = 0, điểm cuối U(0) = 0.
- Với bản tin dài n, thuật toán mã hóa bản tin sẽ lần lượt tìm điểm đầu, điểm cuối cho bản tin dài n = 1, 2,.. là I(n), U(n). Công thức tính lần lượt các điểm này như sau:
  - I(n) = I(n-1) + {U(n-1) – I(n-1)} F(sn-1)
  - U(n) = I(n-1) + {U(n-1) – I(n-1)} F(sn)
  - Với ánh xạ S ={si: i=1,..,q} = {1,..,q}. Tin sn là tin nằm ở cuối bản tin. F(sn-1) là hàm phân bố xác suất của tin ở trước tin sn trong tập tin của nguồn.
  - Chứng minh được I(n) >= I(n-1) và U(n) <= U(n-1). Đoạn của bản tin dài hơn nằm trong đoạn của bản tin ngắn hơn

4.6.2. Mã số học - Thuật toán mã hóa

- Sau khi xác định điểm đầu điểm cuối của đoạn chứa bản tin, điểm biểu diễn bản tin được chọn là điểm có giá trị là phân số nhị phân bé nhất trong đoạn đó. Phần phân số nhị phân của điểm này sẽ được lấy làm mã mã hóa bản tin.
    - Điểm biểu diễn bản tin nằm trong các đoạn biểu diễn các bản tin rút gọn còn 1, 2, 3, .., n tính từ đầu bản tin.
- Với mã số học r trị thì chọn điểm biểu diễn từ mã là điểm có giá trị bằng phân số r phân ngắn nhất trong đoạn và phần phân số này là mã của bản tin.
- Mã số học cho độ dài trung bình của từ mã Hr(S) <= L <= Hr(S) + 2/n. N độ dài bản tin.

4.6.2.Mã số học - thuật toán mã hóa

- Ví dụ: Nguồn S = {a,b,c} có P(S) = {0.7, 0.1, 0.2} và bản tin m = acb
- Hàm phân bố xác suất F(si):
  - si = a -> x = 1
  - si = b -> x = 2
  - si = c -> x = 3



4.6.2.Mã số học - thuật toán mã hóa

- Khi chưa mã hóa (bản tin rỗng) bản tin sẽ là điểm bất kỳ trong đoạn [0,1) hay I(0) =0 và U(0) =1
- Với ký hiệu đầu, bản tin là a:
  - I(1) = I(0) + [U(0) - I(0)]F(si = rỗng) = 0 + (1 -0)0 = 0
  - U(1)= I(0) + [u(0) – I(0)]F(si=a) = 0 + (1 – 0) 0.7 = 0.7
- Với ký hiệu thứ hai, bản tin là ac:
  - I(2) = I(1) + [U(1) – I(1)] F(b) = 0 + (0.7 – 0) 0.8 = 0.56
  - U(2) = I(1) + [U(1) – I(1)]F(c) = 0 + (0.7 -0) 1= 0.7
- Với ký hiệu thứ ba, bản tin là acb:
  - I(3) = I(2) + [U(2) – I(2)] F(a) = 0.56 + (0.7 – 0.56)0.7 = 0.658
  - U(3) = I(2) + {U(2) – I(2)} F(b) = 0.56 + (0.7 -0.56) 0.8 = 0.672
- Chọn điểm biểu diễn bản tin là 0,6640625 có giá trị nhị phân 0.1010101 có số chữ số nhị phân ngắn nhất trong đoạn [0.658, 0.672]. Mã của bản tin sẽ là 1010101.

4.6.2. Mã số học - Thuật toán giải mã

- Mã của bản tin là chuỗi nhị phân nằm trong đoạn biểu diễn bản tin con dài 1,2,.., n tin của bản tin (tính từ đầu bản tin)
- Bắt đầu từ đầu chuối mã nhận được chúng ta sẽ xác định đoạn chứa điểm có giá trị bằng phân số ứng với chuỗi mã nhị phân dài 1, 2,.. chữ số nhị phân. Khi đoạn chứa chuỗi mã nhị phân nằm trong đoạn chứa bản tin con dài n = 1,2,.. ta tách ra được bản tin dàicon n của bản tin.
- Liên tiếp tìm các tin cho đến khi hết tin của bản tin.
- Mã số học không cần truyền bảng mã đến nơi nhận.

4.6.2. Mã số học - Thuật toán giải mã

- Ví dụ: Với nguồn S = (a,b,c) với P(S) = (0.7, 0.1, 0.2) và bản tin m = acb được mã hóa 1010101.
- Với cách mã hóa số học bản tin a nằm trong đoạn [0, 0.7], ac nằm trong đoạn [0.56, 0.7] , acb nằm trong đoạn [0.658, 0.672].
- Chuỗi mã nhận được là 101010
  - Ký hiệu mã đầu 1 ứng với giá trị 0.1 nhị phân, nằm trong đoạn [0.100.., 0.111..] nhị phân hay [0.5,1] chưa cho phép xác định tin đầu tiên vì đây là khoảng chứa ký hiệu đầu có thể là a hoặc b hoặc c.
  - Thêm ký hiệu 10 ứng với giá trị 0.10 nhị phân, nằm trong đoạn [0.1000.., 0.10111..] nhị phân hay [0.5, 0.75], ký hiệu đầu có thể là a hoặc b

4.6.2. Mã số học - Thuật toán giải mã

- Thêm ký hiệu tiếp 101 ứng với giá trị 0.101 nhị phân, nằm trong khoảng [0.10100.., 0.10111..] hay [0.625, 0.75], ký hiệu đầu vẫn có thể a hoặc b
- Thêm ký hiệu tiếp 1010 ứng với giá trị 0.1010, nằm trong khoảng [0.101000.., 0.101011..] hay [0.625, 0.6875]. Đoạn này nằm trong đoạn của tin đầu là a, [0, 0.7] và trong đoạn của tin thứ hai là c, [0.56, 0.7]. Vậy tin đầu của bản tin là a và tin thứ hai là c.
- Tiếp tục 10101 ứng với đoạn [0.65625, 0.6875], 101010 ứng với đoạn [0.65625, 0.67185] không nằm trong đoạn [0.658, 0.672] nên không giải mã được.
- Mã 1010101 ứng với đoan [0.6640625, 0.671875] nằm trong đoạn [0.658, 0.672] ứng với bản tin acb. Vậy bản tin là acb.
- So với mã Huffman, bản tin acb có 6 ký tự. Tuy nhiên nếu phải mở rộng nguồn thì mã số học có hiệu suất mã khá tốt so với Huffman

4.7. Mã hóa nguồn với mã có độ dài cố định

- Source S = {s1,s2,…,sq} with probability P(S) = {P(s1), P(s2),…, P(sq)}
- Codeword has fixed length l → L =l
- Each available combination must have length of l
  - Number of available combination (M) is r^l where r is base of code or number of different code symbols.
  - To satisfy uniquely decodable: minimum number of available combinations equals q where q is number of source symbols.
    - $r^l = q$ → l =$log_r q = H_r(S)max$ →efficient code
      - So, the fixed length code will be efficient code when q = r^l
  - If $r^l \neq q$, we need to choose number of available combination r^l > q
    - According to Shannon, $H_r(S) \leq l \leq H_r(S)$ +1
      - Thus, l = l2 +1 where l2 equals smallest integer greater than to $H_r(S)$
      - Efficiency of code = $\frac{H_r(S)}{l} = \frac{H_r(S)}{l2+1} < 1$

4.7. Mã hóa nguồn với từ mã có độ dài cố định (Cont.)

- To make the efficiency of code can progress to 1, the source needs to be extended
  - S→ S^n: $H_r(S^n)$ = n $H_r(S)$
  - Length of fixed length code for extension source = l_3 +1
    - l_3 smallest integer greater than to $H_r(S^n)$
  - Efficiency of code of extension source $\frac{H_r(S^n)}{l_3+1} < \frac{H_r(S^n)}{H_r(S^n)+1} = \frac{n\,H_r(S)}{n\,H_r(S)+1}$
    - When n→ ∞: efficiency of code → 1

4.8. Mã hóa theo loạt dài (Run-length coding)

- Mã hóa theo loạt dài (RLC) là cách đơn giản và hiệu quả để nén dữ liệu khi một tin xuất hiện liên tiếp nhau trong bản tin.
  - Mã này là thích hợp cho một số loại dữ liệu ảnh, nhưng không luôn tích hợp cho văn bản vì rất hiếm khi một ký tự văn bản xuất hiện 2 lần liên tiếp nhau.
- Để nén một chuỗi tin giống nhau, mã hóa theo loạt dài đơn giản thay chuỗi này bằng 1 tin của chuỗi và tiếp theo là số lần lặp tin này của chuỗi
- Bên giải mã, cần phân biệt mã của số lần lặp và tiến hành phục hồi lại chuỗi tin

4.9. Mã Lempel Ziv

- Mã hóa theo từ điển:
  - Tạo một danh sách các từ và số thứ tự xuất hiện của từ này trong dạng văn bản của bản tin. Danh sách này được gọi là từ điển
  - Chuyển văn bản của bản tin thành chuỗi các chỉ thị chỉ ra điểm vào của từ trong từ điển. Bản tin chuyển thành chuỗi các chỉ số.
  - Khi giải mã từ chuỗi chỉ số là các điểm vào của từ trong từ điển phục hồi lại các từ của bản tin.
  - Mã này phù hợp cho bản tin chuyển về được dạng văn bản.
  - Mã Lempel-Ziv là mã theo từ điển và có nhiều biến dạng. Chúng ta chỉ quan tâm đến hai biến dạng LZ77 và LZ78.

4.9.1. LZ77

- Xây dựng từ theo luật:
    - Mỗi “từ mới” là một “từ cũ” thêm một ký tự tiếp theo trong văn bản
    - Mỗi từ được biểu diễn bởi một bộ ba (m,n,c)
        - n (n >1) là số ký hiệu của chuỗi tìm được trùng với một chuỗi đã xuất hiện trước nó (“từ cũ”.
        - c là ký hiệu tiếp theo chuỗi tìm được.
        - m là số vị trí cần phải quay ngược lại để tìm thấy điểm đầu của chuỗi đã tìm được.
        - Bộ ba đầu tiên là (0,0,c). C là ký tự nguồn đầu tiên. Từ đầu tiên là từ rỗng nằn ở vị trí 0 trong văn bản.

. Văn bản được chuyển thành chuỗi các bộ ba ứng với các từ tìm được.

. Quá trình giải mã:

- Bắt đầu từ bộ ba đầu tiên, dần dần phục hồi lại các từ theo chuỗi các bộ ba nhận được.

4.9.1. LZ77

- Văn b: the_fat_cat_sat_on_the_mat.
- Chuỗi các bộ ba:
  - Bắt đầu từ đầu bản tin : (0,0,t), (0,0,h), (0,0,e),(0,0,_), (0,0,f), (0,0,a), (0,0,t), (0,0,_),(0,0,c), (4,3,s),(4,3,o), (0,0,n), (0,0,_), (19,4,m), (11,2,.)
- Giải mã:
  - (0,0,t)->t, (0,0,h)->h (th), …..→ (0,0,c)->c(the_fat_c), (4,3,s)→(the_fat_cat_s)….

4.9.1. LZ78

- Từ rỗng coi là từ nằm ở vị trí 0 trong từ điển.
- Từ có ví trí 1 tìm được là từ có 1 ký hiệu là ký hiệu đầu của văn bản. Từ này là từ cũ (rỗng lấy thêm một ký tự đầu của vẳn bản). Từ nảy được tách ra khỏi văn bản.
- Tiếp tục tìm các từ có thứ tự tìm được tiếp theo dựa vào nguyên tắc "từ mới" là chuỗi ký hiệu của "từ cũ" và lấy thêm 1 ký hiệu tiếp theo trong văn bản. Từ này được tách khỏi văn bản. Tiếp tục như vậy cho đến khi hết văn bản.
- Mỗi từ được biểu diễn bởi một cặp (i,c)
  - i là số thứ tự tìm thấy từ
  - c là ký tự lấy thêm
- Văn bản là chuỗi các cặp (i,c)

4.9.2. LZ78

- Text: the_fat_cat_sat_on_the_mat.

Item sequence	Code	Item number	Current sequence
t	(0,t)	1	t
h	(0,h)	2	th
e	(0,e)	3	the
_	(0,_)	4	the_
f	(0,f)	5	the_f
a	(0,a)	6	the_fa
t_	(1,_)	7	the_fat_
c	(0,c)	8	the_fat_c
at	(6,t)	9	the_fat_cat
_s	(4,s)	10	the_fat_cat_s
at_	(9,_)	11	the_fat_cat_sat_
o	(0,o)	12	the_fat_cat_sat_o
n	(0,n)	13	the_fat_cat_sat_on
_t	(4,t)	14	the_fat_cat_sat_on_t
he	(2,e)	15	the_fat_cat_sat_on_the
_m	(4,m)	16	the_fat_cat_sat_on_the_m
at.	(9,.)	17	the_fat_cat_sat_on_the_mat.

4.10. Mã nguồn liên tục

- Along with coding, we need to convert the source from analog to digital
  - Sampling
    - After each period of time $T \leq \frac{1}{2}$, one sample is extracted
  - Quantization
    - Each extracted sample value is approximated into one level. The level is integer times of the standard unit
- Obtained results are discrete source. Its output is sequence of levels at time nT
- For continuous source coding, 2 tasks need to be done:
  - Analog to Digital conversion (ADC)
  - Make the number of code symbol used as small as possible (compressing)

4.10. Mã nguồn liên tục

- Mã hóa nguồn cho nguồn liên tục là mã hóa bản tin được nguồn tạo ra với số ký hiệu mã phải sử dụng là tối thiểu
- Mã nguồn tối ưu có độ dài trung bình từ mã L = Hr(S) cho số ký hiệu mã phải sử dụng để mã hóa bản tin là tối thiểu
- Nguồn liên tục được coi là có số tin tạo ra trông 1 đơn vị thời gian bằng số mẫu của nguồn rời rạc tương đương (Ts = 1/(2Fmax))
- Lượng tin trung bình của mỗi mẫu sẽ là Hr(S) nên số ký hiệu mã để mã hóa mỗi mẫu sẽ là L >= Hr(S). Lmin = Hr(s)

4.10. Mã nguồn liên tục

- Mã hóa nguồn cho nguồn liên tục có thể có các phương pháp:
  - Mã hóa theo thời gian là mã hóa bản tin được biểu diễn theo thời gian
  - Mã hóa theo tần số là mã hóa bản tin được biểu diễn theo tần số
  - Mã hóa mô hình nguồn là tìm mô hình toán học cho nguồn và mã hóa các tham số của mô hình

4.10.1. Mã nguồn liên tục theo thời gian

- Mã hóa bản tin là chuyển bản tin thành chuỗi các từ mã hay thành chuỗi ký hiệu mã → Mã hóa nguồn cần thực hiện hai nhiệm vụ:
  - Chuyển đổi bản tin tạo ra từ tương tự sang số (ADC)
  - Giảm số ký hiệu mã đến tối thiểu (hay nén dữ liệu).
- ADC bao gồm 3 thao tác kế tiếp
  - Lấy mẫu
  - Lượng tử hóa
  - mã hóa (thường là mã nhị phân)

4.10. Mã nguồn liên tục theo thời gian

- Hai nhiệm vụ của mã hóa nguồn cho nguồn liên tục có thể thực hiện song song hoặc theo thứ tự:
  - Nén trước khi ADC (nén tương tự):
    - Nén biên độ.
      - Luật nén A: giả sử biên độ x có 0<= |x| <=1. X’ = sign(x){A|x|/(1+lnA)}, khi |x| <=1/A và x’ = sign(x){(1 + ln(A|x|)/(1+lnA)}, khi 1/A < |x| <=1.
    - Nén tần số
      - Tần số Mel: m = 2595lg(1 + f/700). Với f là tần số đo bằng Hz.
  - Song song (nén và các thao tác của ADC xen nhau): Ví dụ PCM kết hợp nén và mã hóa Delta.
    - Lấy mẫu
    - Tính giá trị Delta (đạo hàm rời rạc) để nén chênh lệch biên độ trung bình giữa hai mẫu liên tiếp.
    - Lượng tử hóa Delta (cùng sai số nhưng delta sẽ giảm số giá trị lượng tử)
  - ADC trước khi nén:
    - Đây là thuật toán nén nguồn rời rạc.

4.10. Mã nguồn liên tục theo thời gian

- Với nguồn ảnh thì mỗi ảnh được chia thành các điểm ảnh (Pixel) theo lưới chia tạo thành bởi các đường song song với hai cạnh của ảnh và cách nhau dưới góc nhìn nhỏ hơn hoặc bằng góc phân biệt.
- Thông tin của ảnh là thông tin từ từng điểm của ảnh. Việc đọc thông tin từ các điểm ảnh là thực hiện đọc từng dòng từ trái qua phải và lần lượt từ dòng trên xuống dưới.
- Với ảnh động thì sau khi đọc hết thông tin của 1 ảnh sẽ quay lại đọc từ phần tử góc trên bên trái của ảnh mới
- → Thông tin của ảnh là rời rạc theo thời gian nhưng liên tục theo mức. Trong trường hợp số hóa thì thông tin của mỗi điểm ảnh sẽ được lượng tử hóa thường là 256 mức (1 byte)
- Thông tin của 1 điểm ảnh thường bao gồm 1 thông tin về độ chói (Britgh) với lưới chia theo góc phân biệt 2 phut và 3 thông tin màu với lưới chia theo góc phân biệt 5 phút.

4.10. Mã nguồn liên tục theo thời gian

- Việc nén thông tin của 1 ảnh còn gọi là nén ảnh tĩnh thường là tìm cách mã hóa để tạo ra chuỗi điểm ảnh có thông tin giống nhau liên tiếp và áp dụng mã RLC
- Việc nén ảnh động thường là truyền thông tin 1 ânh đầu và sau đó tìm các thông tin thay đổi của ảnh sau so với ảnh trước và chỉ truyền các thông tin sai khác của ảnh sau.

4.10.1 PCM

PCM encoding example

Level	Code word
0	000
1	001
2	010
3	011
4	100
5	101
6	110
7	111

Levels are encoded using this table

Table: Quantization levels with belonging code words

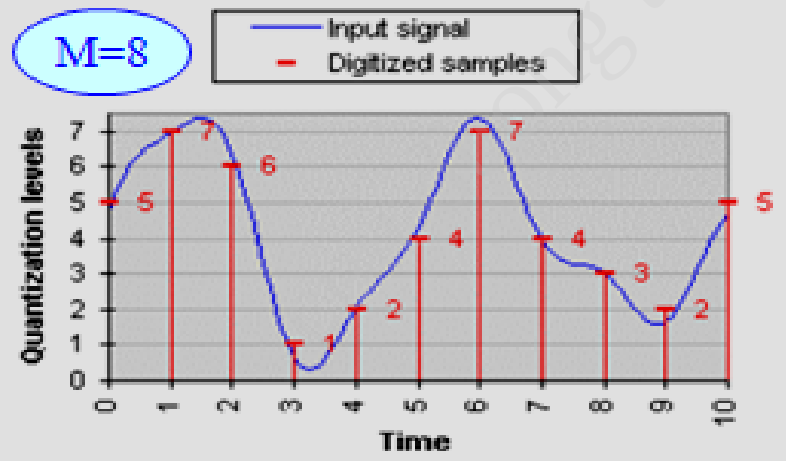


Chart 1. Quantization and digitalization of a signal.

Signal is quantized in 11 time points & 8 quantization segments.

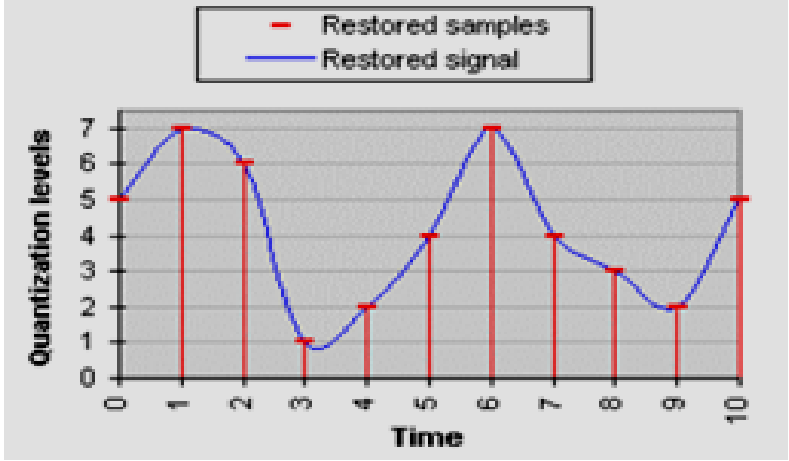


Chart 2. Process of restoring a signal.

PCM encoded signal in binary form:

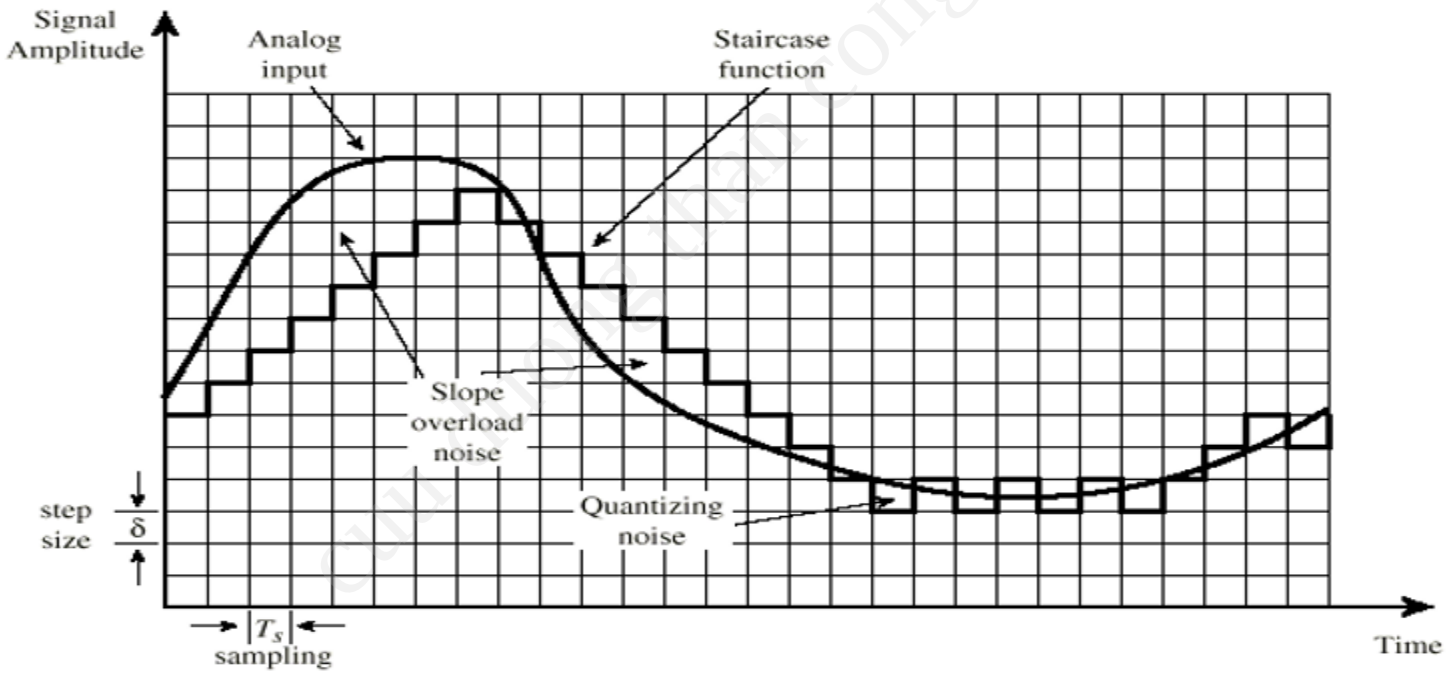
101 111 110 001 010 100 111 100 011 010 101

Total of **33** bits were used to encode a signal

4.10.1. PCM

- Để giảm số bit dùng mã hóa PCM áp dụng thêm một luật nén:
  - Nén biên độ trước PCM làm cho dải động Xmax – Xmin giảm dẫn đến số mức lượng tử M = (Xmax – Xmin)/ Vref +1 giảm → số bit mã hóa mỗi mẫu n = log2(M) giảm
  - Tính giá trị Delta = mẫu(i) – mẫu(i-1). Delta là đạo hàm rời rạc của bản tin nên dải động của Delta sẽ bé hơn

4.10.1. Mã hóa delta

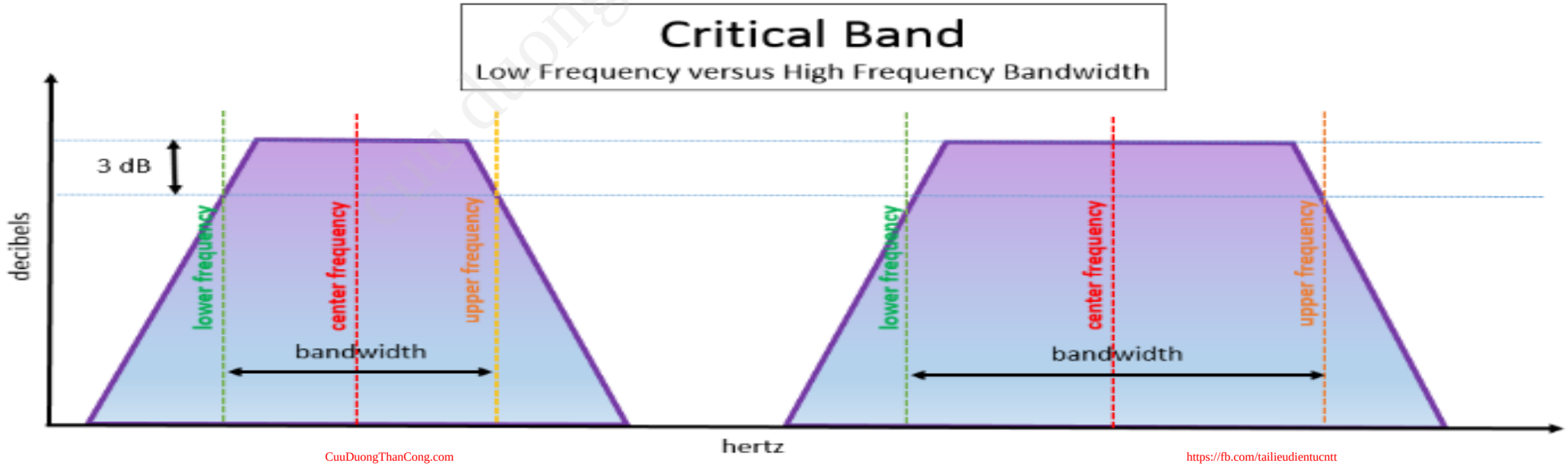


4.10.2. Mã hóa theo tần số

- Mã hóa nguồn liên tục theo tần số sẽ mã hóa phổ bản tin được tạo ra.
- Thông thường các giải pháp mã hóa tần số là mã hóa tham số → ước lượng tham số của phổ rồi mã hóa nó
- Do nguồn nói chung có mô hình là quá trình ngẫu nhiên ergodic nên tham số biến thiên chậm → lấy mẫu thưa hơn và dải động nhỏ
- Hai kỹ thuật hay được sử dụng là mã hóa theo dải băng tần con và mã hóa theo các thành phần cơ bản (ví dụ mã hóa formant)

4.10.2. Mã hóa theo dải băng tần con

- Mã hóa theo băng tần con dựa vào khái niệm băng tần tới hạn (Critical Band) của các cơ quan cảm nhận: cơ quan cảm nhận không phân biệt được độ cao của từng tần sô riêng mà theo từng dải tần số == Băng tới hạn
- Ví dụ tai người phân biệt các băng tần số (giống như sử dụng bộ lọc thông dải) có tần số trung tâm và độ rộng của băng tăng dần theo tần số
- .



4.10.2. Mã hóa theo dải băng tần con

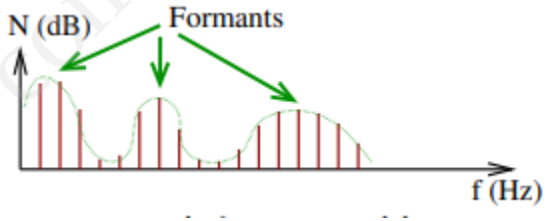
- Dải tần số tai người nghe tốt sẽ có 24 băng tới hạn, mỗi băng gọi là 1 Bark (hay tính theo độ đo Bark)
- Mã hóa theo dải băng tần con sẽ sử dụng một hệ thống các mạch lọc thông dải có tham số tương ứng với tham số của các băng tới hạn. Các đầu ra của các mạch lọc là biên độ trung bình của tín hiệu của mỗi Băng con.

Bên giải mã sẽ sử dụng hệ thống mạch lọc giống bên mã hóa với đầu vào là nhiễu trắng. Hệ số truyền đạt của các mạch lọc này là đầu ra của các bộ lọc mã hóa. MP3 là loại mã này cho âm thanh.

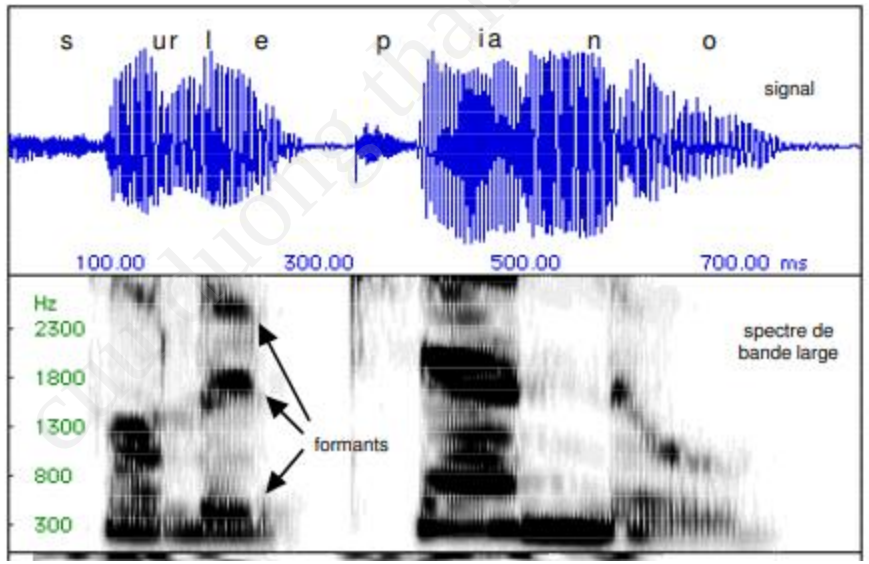
Critical Band (Bark)	Center Frequency (Hz)	Bandwidth (Hz)
1	50	100
2	150	100
3	250	100
4	350	100
5	450	110
6	570	120
7	700	140
8	840	150
9	1000	160
10	1170	190
11	1370	210
12	1600	240
13	1850	280
14	2150	320
15	2500	380
16	2900	450
17	3400	550
18	4000	700
19	4800	900
20	5800	1100
21	7000	1300
22	8500	1800
23	10500	2500
24	13500	3500

4.10.2. Mã hóa theo formant

- Hình bên là khái niệm formant trong phổ



- Hình dưới là phổ đồ (spectograme) của đoạn âm thanh



4.10.2. Mã hóa theo mô hình nguồn

- Phương pháp mã hóa theo mô hình nguồn là phương pháp tìm mô hình toán học cho nguồn từ bản tin được tạo ra
- Phương pháp mã hóa theo mô hình nguồn được ứng dụng nhiều trong thực tế xử lý âm thanh là mã hóa dự đoán tuyến tính (Linear Predictive Coding).
- Phương pháp mã hóa dự đoán tuyến tính coi mỗi mẫu tín được tạo ra s[n] là tổ hợp tuyến tính của các mẫu tạo ra trước nó s[n – k]

$$s[n] = \sum_{k=1}^{p} a_k s[n-k] + e[n]$$

- Ở đây ak là các hệ số cần tìm của bộ dự đoán cấp p, e[n] là sai số dự đoán mẫu s[n], p Nếu dự đoán đúng thì sai số e[n] =0. Hay phương trình trên khi e[n] = 0 là phương trình sai phân của 1 nguồn tạo ra các mẫu tin s[n]

4.10.2. Mã hóa theo mô hình nguồn

- Phương trình trên có thể coi là phương trình sai phân của 1 hệ tuyến tính bất biến với đầu vào là e[n] và đầu ra s[n]. Hàm truyền của hệ trong không gian z:

$$\frac{S(z)}{E(z)} = \frac{1}{1 - \sum_{k=1}^{p} a_k z^{-k}} = \frac{1}{A(z)}$$

- Đây là mô hình của hệ chỉ có các điểm cực hay hệ tự hồi quy (AutoRegressive)
- Để hệ trên trở thành mô hình nguồn ta cần tìm các hệ số ak sao cho e[n] -> 0 (đạt giá trị cực tiểu)

4.10.2. Mã hóa theo mô hình nguồn

- Chọn các ak để cực tiểu hóa e[n] có thể đạt được bằng cách cực tiểu hóa năng lượng sai số khi chọn các ak theo phương trình

$$\sum_n e^2[n] = \sum_n \left(s[n] - \sum_{k=1}^{p} a_k s[n-k] \right)^2$$

- Đạo hàm phương trình trên theo từng hệ số ak và tìm giá trị ak cho phương trình đạo hàm bằng 0 sẽ là hệ p phương trình với các hệ số ak cho năng lượng tối thiểu cho phép giải ra các hệ số ak và sai số dự đoán e[n]
- Kết quả là tập hệ số ak của mô hình nguồn và sai số dự đoán e[n] tối thiểu tìm được là đầu vào của mô hình. Đây là các tham số trong mã hóa mô hình nguồn.
- Bên giải mã sẽ là hệ tự hồi quy với tập hệ số ak và đầu vào là e[n].

Bài tập

- Bài 4: Giả sử nguồn tin chỉ tao một bản tin có nội dung là ‘công nghệ thông tin’ viết ở dạng tiếng Việt không dấu, không phân biệt chữ thường chữ hoa và không có dấu cách giữa các từ. Mỗi ký tự trong bản tin là 1 tin được tạo ra từ nguồn. Xác suất xuất hiện của mỗi tin là tần suất xuất hiện của nó trong bản tin.
  - a. Hãy mã hóa bản tin trên theo mã Huffman với cơ số mã r = 4.
  - b. Tính hiệu suất của mã
  - c. Cần mở rông nguồn bao nhiêu lần ở hiệu suất mã đạt ít nhất 98%
- Bài 5: Hãy mã hóa bản tin ở câu 4 bằng mã LZ77 và bằng mã LZ78