ELEC S347F Multimedia Technologies,
Spring Term, 2021
Tutorial 02 Solution

Question 1: Consider a 3-symbol source data sequence: {A, B, C, B, B, A, B, C, A, B}. Encode the source using the below codes respectively.

(a) Shannon-Fano code
(b) Huffman code
(c) 3-bit Tunstall code
(d) 3-bit Tunstall/Huffman code

For each code,

(i) Show the code table.
(ii) Show the encoded string and its length.
(iii) Compute the expected redundancy and efficiency for the code.
(iv) Compute the effective redundancy and efficiency for this source.

Symbol occurrence probabilities: $p(A) = 0.3$, $p(B) = 0.5$, $p(C) = 0.2$.

Entropy H = $-0.3\log_2(0.3)-0.5\log_2(0.5)-0.2\log_2(0.2)$ = 1.485475 bits/symbol

(a) Shannon-Fano Code

{A: 0.3, B: 0.5, C: 0.2}

{B: 0.5, A: 0.3, C: 0.2}

{{B: 0.5}, {A: 0.3, C: 0.2}}

{{B: 0.5}, {{A: 0.3}, {C: 0.2}}

Symbol	Codeword	Probability
A	10	0.3
B	0	0.5
C	11	0.2

Code Rate r = Average length per codeword / average length per symbol

= (0.3x2 + 0.5x1 + 0.2x2)/1 = 1.5 bits/symbol

Code Redundancy R = Code Rate – Entropy = 0.014525 bits/symbol

Code Efficiency = Entropy / Code Rate = 0.99032

Encoded sequence = 10 0 11 0 0 10 0 11 10 0 (length = 15 bits)

(b) Huffman Code

Build the Huffman tree (the tree diagram is omitted here)

{B: 0.5, A: 0.3, C: 0.2}

{B:0.5, (AC): 0.5}

{(B(AC)): 1.0}

Symbol	Codeword	Probability
A	10	0.3
B	0	0.5
C	11	0.2

Code Rate r = (0.3x2 + 0.5x1 + 0.2x2)/1 = 1.5 bits/symbol

Code Redundancy R = Code Rate – Entropy = 0.014525 bits/symbol

Code Efficiency = Entropy / Code Rate = 0.99032

Encoded sequence = 10 0 11 0 0 10 0 11 10 0 (length = 15 bits)

(c) 3-bit Tunstall Code

Symbol	Probability
A	0.3
B	0.5
C	0.2

Symbol	Probability
A	0.3
~~B~~	~~0.5~~
C	0.2
BA	0.15
BB	0.25
BC	0.10

Symbol	Probability	Codeword
~~A~~	~~0.3~~	
~~B~~	~~0.5~~	
C	0.2	000
BA	0.15	001
BB	0.25	010
BC	0.10	011
AA	0.09	100
AB	0.15	101
AC	0.06	110

Code Rate = 3/(0.2x1 + 0.15x2 + 0.25x2 + 0.10x2 + 0.09x2 + 0.15x2 + 0.06x2) = 1.66667 bits/symbol

Redundancy = 0.18119 bits/symbol

Efficiency = 0.891285

Encoded sequence: (length = 18 bits)

A B	C	B B	A B	C	A B
101	000	010	101	000	101

Or assign the unused codeword '111' to symbol 'A':

A	B C	B B	A	B C	A B
111	011	010	111	011	101

Note 1: effective code rate = 18 bits/10 symbols = 1.8 bits/symbol (> expected code rate since the occurrence probabilities of the combined symbols are different from expected)

Note 2: both length are 18 bits long. It is non-uniquely encodable

(d) 3-bit Tunstall/Huffman Code

Based on the result of Tunstall coding to build the Huffman tree:

{BB: 0.25, C: 0.2, AB: 0.15, BA: 0.15, BC: 0.10, AA: 0.09, AC: 0.06}

{BB: 0.25, C: 0.2, (AA, AC): 0.15, AB: 0.15, BA: 0.15, BC: 0.10}

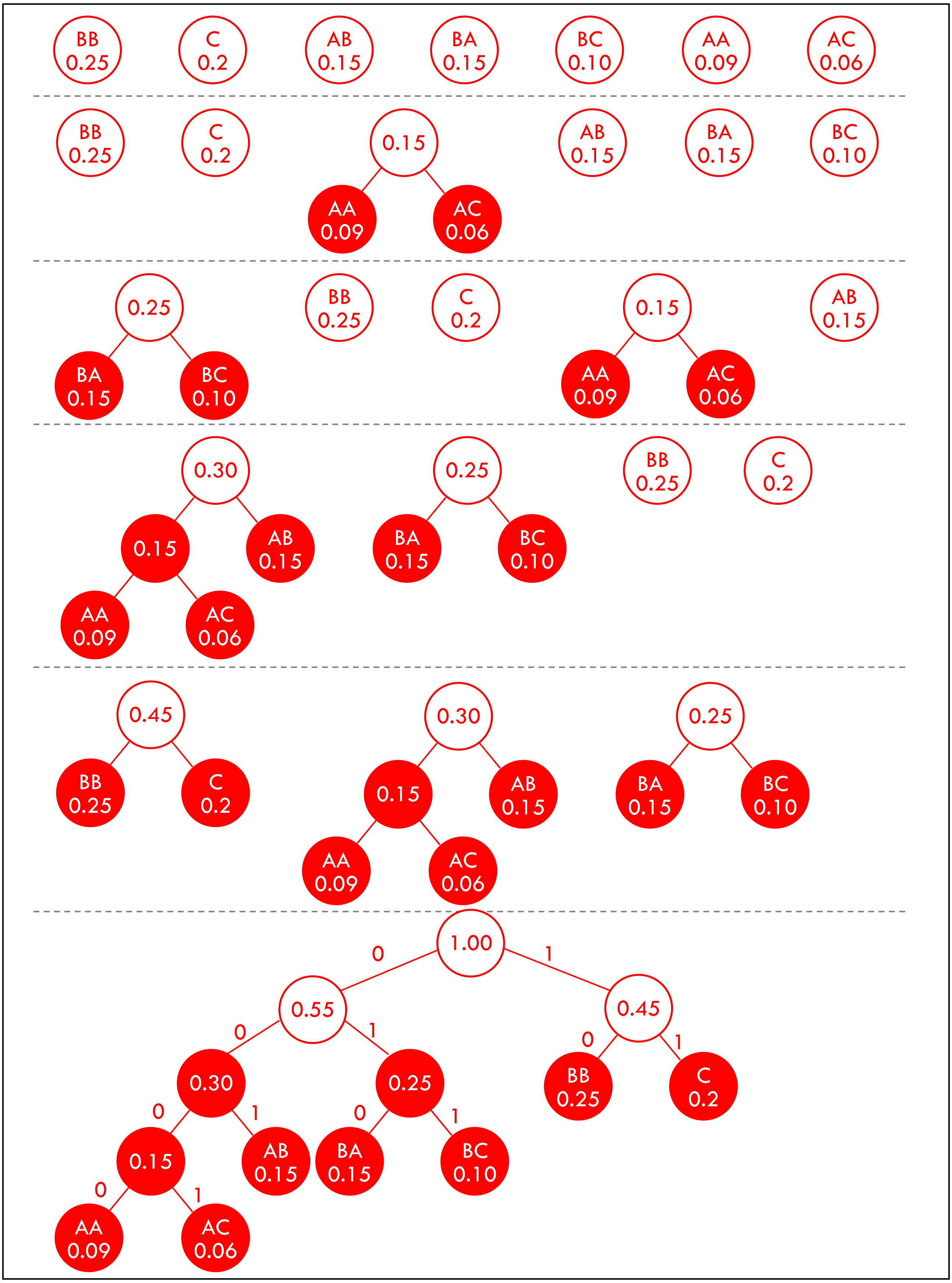
{(BA, BC): 0.25, BB: 0.25, C: 0.2, (AA, AC): 0.15, AB: 0.15}

{(BB, C): 0.45, ((AA, AC), AB): 0.30, (BA, BC): 0.25}

{(((AA, AC), AB), (BA, BC): 0.55, (BB, C): 0.45}

{((((AA, AC), AB), (BA, BC), (BB, C)): 1.00}

Symbol	Probability						
BB (10)	0.25	0.25	[0.25]	[0.30]	[0.45]	[0.55]	1.0
C (11)	0.20	0.20	0.25	0.25	0.30	0.45	
AB (001)	0.15	[0.15]	0.20	0.25	0.25		
BA (010)	0.15	0.15	0.15	0.20			
BC (011)	0.10	0.15	0.15				
AA (0000)	0.09	0.10					
AC (0001)	0.06						


BB 0.25
C 0.2
AB 0.15
BA 0.15
BC 0.10
AA 0.09
AC 0.06
BB 0.25
C 0.2
0.15
AA 0.09
AC 0.06
AB 0.15
BA 0.15
BC 0.10
0.25
BA 0.15
BC 0.10
BB 0.25
C 0.2
0.15
AA 0.09
AC 0.06
AB 0.15
0.30
0.15
AB 0.15
AA 0.09
AC 0.06
0.25
BA 0.15
BC 0.10
BB 0.25
C 0.2
0.45
BB 0.25
C 0.2
0.30
0.15
AB 0.15
AA 0.09
AC 0.06
0.25
BA 0.15
BC 0.10
1.00
0
1
0.55
0.45
0
1
0
1
0.30
0.25
BB 0.25
C 0.2
0
1
0
1
0.15
AB 0.15
BA 0.15
BC 0.10
0
1
AA 0.09
AC 0.06

Symbol	Probability	Codeword
C	0.2	11
BA	0.15	010
BB	0.25	10
BC	0.10	011
AA	0.09	0000
AB	0.15	001
AC	0.06	0001

Code Rate = Average length of codewords / average length of symbols

= (0.2x2 + 0.15x3 + 0.25x2 + 0.1x3 + 0.09x4 + 0.15x3 + 0.06x4) / (0.2x1 + 0.15x2 + 0.25x2 + 0.10x2 + 0.09x2 + 0.15x2 + 0.06x2) = 1.5 bits/symbol

Redundancy = 0.014525 bits/symbol

Efficiency = 0.99032

Encoded sequence = 001 11 10 001 11 001 (AB, C, BB, AB, C, AB) (length = 15 bits)

Effective code rate = 15 bits / 10 symbol = 1.5 bits/symbol (= expected code rate)

Question 2:

(a) Consider a 2-symbol source data: {A, B, B, A, B}. Encode the source using arithmetic coding. Fill your answer in the below table.

Symbol	Range of A	Range of B	Encoded Range	Encoded Interval
A	[0, 0.4)	[0.4, 1)	[0, 0.4)	0.4
B	[0, 0.16)	[0.16, 0.4)	[0.16, 0.4)	0.24
B	[0.16, 0.256)	[0.256, 0.4)	[0.256, 0.4)	0.144
A	[0.256, 0.3136)	[0.3136, 0.4)	[0.256, 0.3136)	0.0576
B	[0.256, 0.27904)	[0.27904, 0.3136)	[0.27904, 0.3136)	0.03456

P(A) = 0.4, P(B) = 0.6

Entropy H = $-(0.4)\log_2(0.4)-(0.6)\log_2(0.6)$ = 0.970951 bit/symbol

$[0.27904_{(10)}, 0.3136_{(10)}) = [0.0100011.._{(2)}, 0.0101000.._{(2)})$

The optimal encoded number is 0.28125 = $2^{-2} + 2^{-4} + 2^{-5}$= 0.01001_2.

So the encoded sequence = 01001

(b) Given a source with the same probability distribution model in (a) and length equals 5 is arithmetic coded as 101011. Decode the string by filling your answer in the below table.

Symbol	Range of A	Range of B	Decoded Symbol	Decoded Interval
1	[0, 0.4)	[0.4, 1)	B	0.6
2	[0.4, 0.64)	[0.64, 1)	B	0.36
3	[0.64, 0.784)	[0.784, 1)	A	0.144
4	[0.64, 0.6976)	[0.6976, 0.784)	A	0.0576
5	[0.64, 0.66304)	[0.66304, 0.6976)	B	0.03456

Encoded string = 101011. i.e. the encoded number $x = 0.101011_2 = 0.671875$

Decoded sequence = B, B, A, A, B

Code rate $r = 6/5 = 1.2$

Code Efficiency $E = H/r = 0.809125$

Code Redundancy $R = r - H = 0.229049$ bit/symbol

Question 3: Given the initial dictionary (code table) of a 3-bit LZW code as follows

Symbol	Codeword
a	000
b	001

If the source message is {a, b, b, b, a, b, b, b, a},

(a) Determine the encoded sequence.

(b) Determine the final dictionary (code table) and its size.

W	K	WK	Output
-	a	a	-
a	b	ab	000
b	b	bb	001
b	b	bb	-
bb	a	bba	011
a	b	ab	-
ab	b	abb	010
b	b	bb	-

Symbol	Codeword
a	000
b	001
ab	010
bb	011
bba	100
abb	101

bb	a	bba	-
bba	-	-	100

(a) The encoded process is shown in the above table and the encoded sequence is 000 001 011 010 100 (Total bits required = 15 bits)

(b) The final dictionary is shown in the above table and it has 6 entries.

Question 4: Given the initial dictionary (code table) of a 3-bit LZW code as follows:

Symbol	Codeword
a	000
b	001

If the received LZW encoded sequence is 000 001 011 010 100,

(a) Determine the original message.

(b) Determine the final dictionary (code table) and its size.

W	K	sym(K)	W+sym(K)[0]	Output
-	000	a	a	a
a	001	b	ab	b
b	011	? (bb)	bb	bb
bb	010	ab	bba	ab
ab	100	bba	abb	bba
bba				

Symbol	Codeword
a	000
b	001
ab	010
bb	011
bba	100
abb	101

(a) The decode process is shown in the above table.

The original message = a b bb ab bba

(b) The final dictionary is shown in the above table and it has 6 entries.

Comment: in general, decoding requires fewer iterations. This is an example of asymmetric encoding/decoding.