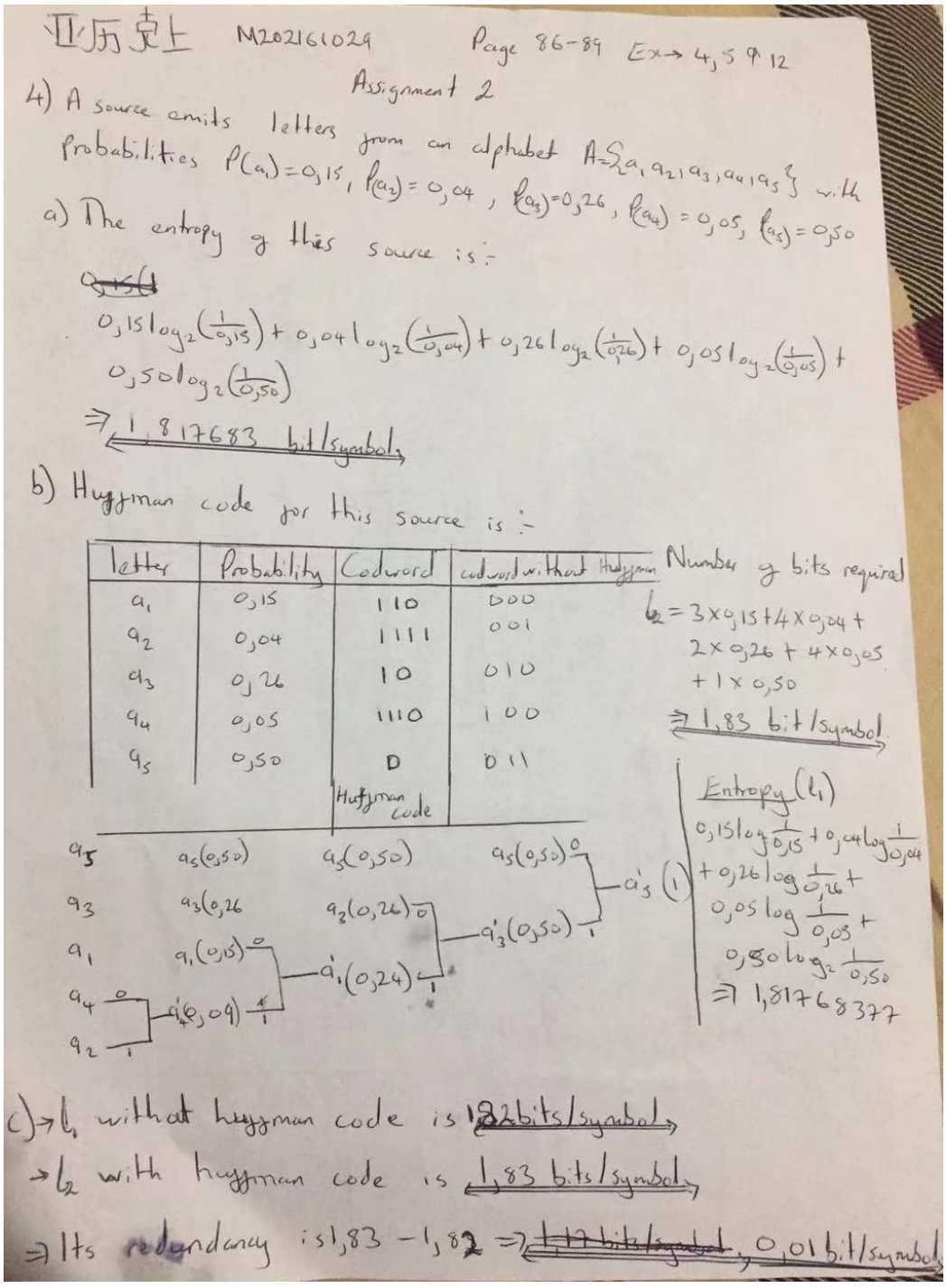
Alec Mabhiza Chirawu - 亚历克上 - M202161029

Advanced Coding
Assignment 2



亚历克上 M202161029 Page 86-89 Ex→ 4, 5 & 12

Assignment 2

4) A source emits letters from an alphabet $A=\{a_1, a_2, a_3, a_4, a_5\}$ with probabilities $P(a_1)=0,15$, $P(a_2)=0,04$, $P(a_3)=0,26$, $P(a_4)=0,05$, $P(a_5)=0,50$

a) The entropy of this source is:-

$$0,15\log_2\left(\frac{1}{0,15}\right)+0,04\log_2\left(\frac{1}{0,04}\right)+0,26\log_2\left(\frac{1}{0,26}\right)+0,05\log_2\left(\frac{1}{0,05}\right)+0,50\log_2\left(\frac{1}{0,50}\right)$$

$\Rightarrow 1,817683$ bit/symbol

b) Huffman code for this source is:-

letter	Probability	Codeword	codeword without Huffman
a_1	0,15	110	000
a_2	0,04	1111	001
a_3	0,26	10	010
a_4	0,05	1110	100
a_5	0,50	0	011
		Huffman code	

Number of bits required

$l_2 = 3\times0,15+4\times0,04+2\times0,26+4\times0,05+1\times0,50$

$\Rightarrow 1,83$ bit/symbol

Entropy(l_1)

$0,15\log\frac{1}{0,15}+0,04\log\frac{1}{0,04}+0,26\log\frac{1}{0,26}+0,05\log\frac{1}{0,05}+0,50\log_2\frac{1}{0,50}$

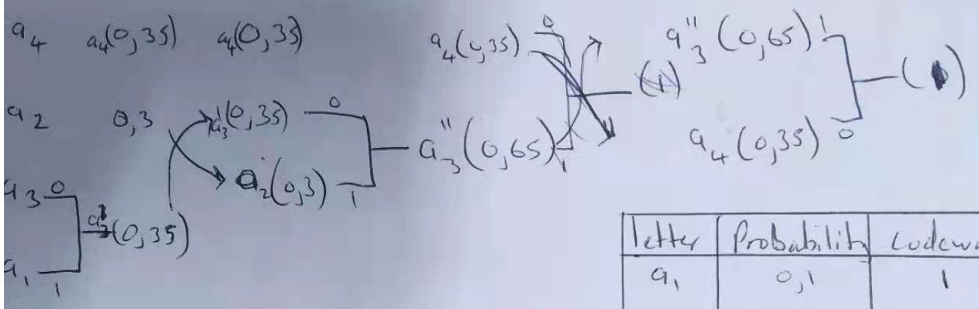
$\Rightarrow 1,81768377$

c) $\rightarrow l_1$ without huffman code is 1,82 bits/symbol

$\rightarrow l_2$ with huffman code is 1,83 bits/symbol

$\Rightarrow$ Its redundancy is $1,83-1,82 \Rightarrow$ 0,01 bit/symbol

a) The first procedure outlined in this chapter



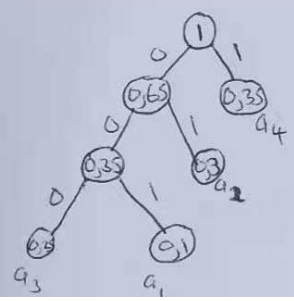
letter	Probability	codeword
q_1	0,1	1
q_2	0,3	01
q_3	0,25	000
q_4	0,35	0

5) An alphabet $A = \{a_1, a_2, a_3, a_4\}$ with probabilities $p(a_1) = 0,1$, $p(a_2) = 0,3$ $p(a_3) = 0,25$ and $p(a_4) = 0,35$

b) Finding Huffman coding using the minimum procedure outlined in this chapter (Tree method)

∴ Rules of the tree structure :- (i) when Next Element > Previous Element ⇒ left
(ii) when NExt Element < PE ⇒ Right
(iii) when NE = PE ⇒ Left

a_4 a_2 a_3 a_1
0,35 0,3 0,25 0,1



letter	Probability	codeword Huffman code
q_1	0,1	1
q_2	0,3	011
q_3	0,25	000
q_4	0,35	1

Comment

- The first procedure and The tree (the minimum) huffman codes have the same sequence as it the lowest probabilities are added first.
- The first procedure swap the new value if its greater than the upper value whereas minimum procedure has 3 rules which are (i) when Next element > Previous Element ⇒ left
  (ii) when Next element < Previous Element ⇒ right
  (iii) when Next element = Previous Element ⇒ left

→ They both gives the same results (codeword)