

Math 320 Homework 2.6

Chris Rytting

October 6, 2015

2.31

The fact that f is bijective implies that every element of the source alphabet S maps to a unique codeword in C , and that every codeword in C is mapped to by an element of S . That f is instantaneous means that no codeword is a prefix of any other codeword, and that a codeword can be decoded as soon as a codeword is received. Assume to the contrary that it is not uniquely decipherable. Note that the only time one could actually decode any part of a message is if they thought they had found an entire codeword. The only time this might cause confusion is if this codeword were the prefix of a different codeword, for if the intended codeword were finished and decoded, there would be no confusion. However, the definition of instantaneous implies that this is not the case, and we have the desired result by contradiction.

2.32

Let $S = a, b, c$ and $C = 0, 01, 011$. We have that 'CAB' maps to 011001. If we try to interpret the message after receiving the first message, though, we will get an a in the first spot, so it is not instantaneous. However, the mapping is unique and if we work from right to left, once the whole message is received, then we will only get a certain message back.

2.33

Consider the following probability distribution $P(a) = .2, P(b) = .2, P(c) = .2, P(d) = .5, P(e) = .5, P(f) = .9$. In this case, we can either start with a and b or we can start with b and c . Even if we tried to do everything else the same afterwards, in one case there would be an a and a b in the bottom level of the tree or there would be a b and a c , meaning that we couldn't permute either tree to resemble the other by switching 0 and 1.

2.34

$$a : 001, b : 000, c : 1, d : 010, e : 011$$

2.35

$$\begin{aligned}P(,) = 0.027, P(.) = 0.027, P(,"") = 0.189, P(t) = 0.081, P(h) = 0.081, P(e) = 0.135, P(a) = 0.027 \\ P(r) = 0.108, P(d) = 0.027, P(i) = 0.081, P(w) = 0.027, P(o) = 0.027, P(k) = 0.054 \\ P(l) = 0.027, P(u) = 0.027, P(c) = 0.027, P(g) = 0.027\end{aligned}$$

Reordering and performing a Huffman encoding on this set yields an alphabet

$$\begin{aligned}t &= 0000 \\ h &= 0010 \\ e &= 010 \\ a &= 01110 \\ r &= 0001 \\ d &= 10000 \\ "" &= 11 \\ i &= 0011 \\ w &= 10101 \\ o &= 10011 \\ k &= 0110 \\ l &= 10010 \\ u &= 10100 \\ c &= 01111 \\ g &= 10001\end{aligned}$$

To compress the string, simply replace each letter in the string with the given binary representation.

There will be 140 bits for the Huffman encoding here described (multiply the number of instances of each letter by the number of bits in each letter (4 for t , 3 for e , etc.), and then sum these products), and $37 * 8 = 296$ bits for ASCII.,