

Anmerkung zu 1: Die Berechnungen wurden mithilfe der Polynomdivision durchgeführt, wie auf den Folien beschrieben, wurden aber hier weglassen, da sich die Polynomdivision nur sehr schwer in Word/Excel o.ä. darstellen lässt.

1a. u^4 + u^3 +1 = 11001

Primitiv, da kleinstes Polynom das sich durch g1(u) teilen lässt u^15 +1 ist, Grad von g1(u) aber 4 ist  
-> 2(^4)-1=15

1b. u^4 + u^2 + u + 1= 10111

Nicht primitiv, da sich das Polynom P(u)=u^7 +1 restlos durch g2(u) teilen lässt, aber Grad(g2(u))=4 gilt und somit grad des ersten restlos durch g(u) teilbaren Polynom von der Form u^q +1 vom Grad 15=2(^4)-1 sein muss.

1c. u^4 + u^3 + u + 1= 11011

Nicht primitiv, da sich das Polynom P(u)=u^6 +1 restlos durch g2(u) teilen lässt, aber Grad(g2(u))=4 gilt und somit grad des ersten restlos durch g(u) teilbaren Polynom von der Form u^q +1 vom Grad 15=2(^4)-1 sein muss.

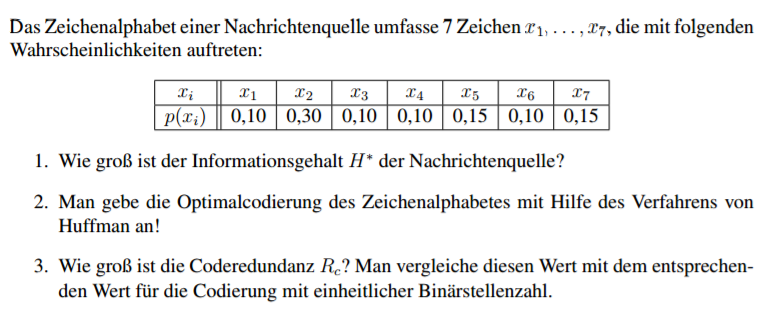
3. Falls das Polynom eine Nullstelle bei -1 hat ist es durch (u+1) teilbar, da wenn ja, sich das Polynom als P(u)=(u+1)\*Q(u) darstellen lässt und für P(-1)=0\*Q(-1)=0 gilt.

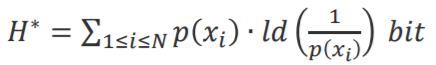
🡪 Anwendung dieser Regel bei 2:

1. g(-1)=(-1)^4+(-1)^3+1=1 -> nicht durch (u+1) teilbar

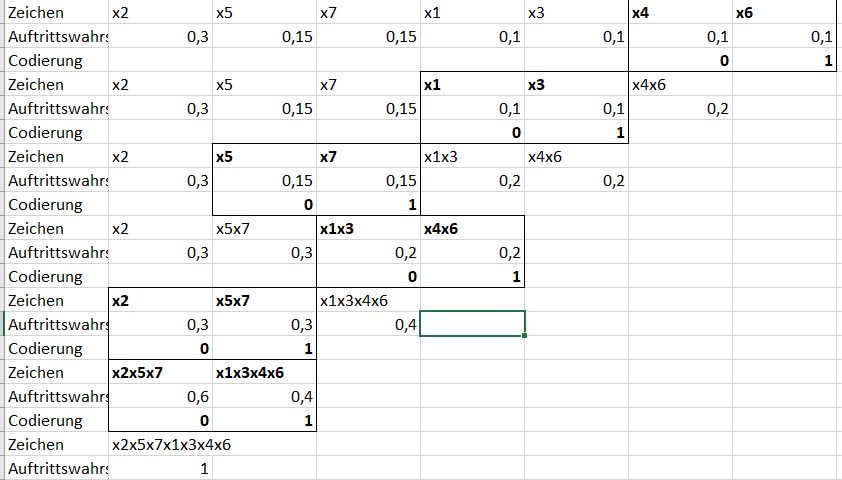
2. g(-1)=1+1-1+1=2 -> nicht durch (u+1) teilbar

3.g(-1)=1-1-1+1=0 -> durch (u+1) teilbar



1. = 4\*(0,1\*ld(10))+2\*(0,15\*ld(6.6666666666…))+(0,3\*ld(3,3333333..))=2,11 2,67

2. Sortierung nach fallender Auftrittswahrscheinlichkeit ergibt folgende Reihenfolge:  
x2 x5 x7 x1 x3 x4 x6  
-> Codierung:



X2=00

X5=010

X7=011

X1=100

X3=101

X4=110

X6=111

3. Berechnung = 2,7 bit

= 2,7bit-2,67bit=0,03 bit

Bei einheitlicher Codierung mit Binärstellenzahl würde jedes Codewort aus 3 bit bestehen, da wir 7 Codewörter haben und somit da 2^3=8 für die eindeutige Darstellung dieser min. 3 bit brauchen.

Damit wäre die Redundanz Rc = 3bit – 2,67bit= 0,33 bit

D.h. die Huffmankodierung ist hier im Vergleich zur simplen Binärkodierung um 0,3 bit pro Codewort effizienter.