## Small Alphabet

- What will happen if the items set for the compression is small?
- Example of 77.49MB of bitmaps:

| | n | Huffman (MB) | Arithmetic (MB) | increase (%) |
|---|---|---|---|---|
| each bit | 2 | 77.49 | 9.64 | 703 |
| k-blocks | 256 | 13.70 | 7.70 | 78 |




## Arithmetic evaluation

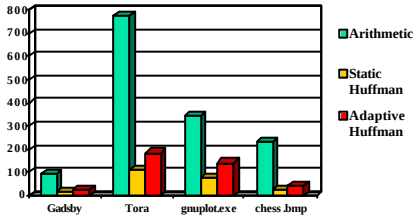
- Average codeword length in bits in Huffman and Arithmetic:

| | Huffman | Arithmetic | Huffman cost (%) |
|---|---|---|---|
| English | 4.1854 | 4.1603 | 0.6 |
| Finnish | 4.0448 | 4.0134 | 0.8 |
| French | 4.0003 | 4.0376 | 0.9 |
| German | 4.1475 | 4.1129 | 0.8 |
| Hebrew | 4.2851 | 4.2490 | 0.8 |
| Italian | 4.0000 | 3.9725 | 0.7 |
| Portuguese | 4.0100 | 3.9864 | 0.6 |
| Spanish | 4.0469 | 4.0230 | 0.6 |
| Russian | 4.4704 | 4.4425 | 0.6 |
| English-2 | 7.4446 | 7.4158 | 0.4 |
| Hebrew-2 | 8.0370 | 8.0085 | 0.4 |




## Time considerations

- Decoding time in seconds:

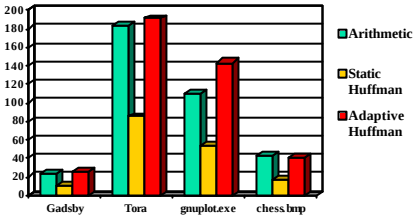






## Time considerations

- Encoding time in seconds:







## An example

- The items are:
  - **a-10%**
  - **i-20%**
  - **r-30%**
  - **y-40%**
- A Huffman tree for these item:
  - y - 0
  - r - 10
  - a - 110
  - i – 111
- The string which should be encoded is: "ii".




## Quasi Arithmetic Coding

- A combination of Huffman coding and Arithmetic coding.
- Algorithm:
  - Compress data by Huffman (Or other prefix codes).
  - Compress the bits by Arithmetic coding.

## An example (Cont.)

- The bit string for "ii" is: 111111.
- $[0,1) \rightarrow [8/19,1) \rightarrow [240/361,1) \rightarrow$

$\rightarrow [5528/6859,1) \rightarrow [115680/130321,1) \rightarrow$

$\rightarrow [2315048/2476099,1)$

$\rightarrow [45274320/47045881,1)$



## An example (Cont.)

- ones - 0.3*1+0.1*2+0.2*3=1.1
- zeros - 0.4*1+0.3*1+0.1*1=0.8
- Ratio is 8:11
- Arithmetic code for bits:
  - $[0,8/19)$
  - $[8/19,1)$



## Advantages of Quasi Arithmetic

- Just two intervals (for 1 and 0) so can be implemented by tables instead of floating point divisions.
- These table can be implemented even in hardware.

$\Rightarrow$ Faster than Arithmetic coding but has the same efficiency.



## An example (Cont.)

- The shortest binary fraction in this interval is 0.11111 which is just 5 bits.
- Huffman represents "i" by 3 bits which is too much since $3=-\log_2 0.125$ while the probability of "i" is 0.2.
- $0.2*0.2=0.04$ so the content of data in this string is $-\log_2 0.04$ which is circa 4.643856 so 5 bits are the optimal compression of this string.



## Theorem - Proof

- The derivative of the entropy is:

$$H' = [-\sum_{i=0}^{1} P_i \log_2 P_i]' = [-P\log_2 P - (1-P)\log_2(1-P)]' =$$

$$= [\frac{-P\ln P}{\ln 2} - \frac{(1-P)\ln(1-P)}{\ln 2}]' = \frac{-P}{P\ln 2} + \frac{-\ln P}{\ln 2} - \frac{(1-P)(-1)}{\ln 2(1-P)} - \frac{-\ln(1-P)}{\ln 2} =$$

$$= \frac{-1}{\ln 2} - \log_2 P + \frac{1}{\ln 2} + \log_2(1-P) = \log_2(\frac{1-P}{P})$$



## Theorem

- The second step of the Quasi Arithmetic Coding will give the worst improvement when the 1s and the 0s have exactly the same probability - 0.5
- In other words: The entropy of a binary string will be maximal, if the 1s and the 0s have exactly the same probability - 0.5
- When the ratio between the 1s and the 0s grows, the entropy will be diminished.

## Theorem - Proof (Cont.)

- The second derivative is:

$$[\log_2(1-P)-\log_2 P]'=[\frac{\ln(1-P)}{\ln 2}-\frac{\ln P}{\ln 2}]'=-\frac{1}{(1-P)\ln 2}-\frac{1}{P\ln 2}$$

- When P=1/2:

$$-\frac{2}{\ln 2}-\frac{2}{\ln 2}<0$$

- So 1/2 is a maximum point.

## Theorem - Proof (Cont.)

$$\log_2(\frac{1-P}{P})=0$$

$$\frac{1-P}{P}=1$$

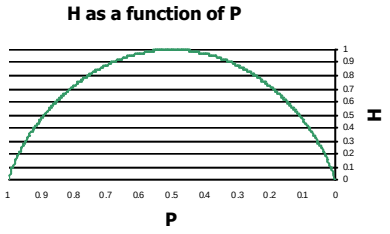
$$1-P=P$$

$$P=\frac{1}{2}$$

- So there is an extremum point when P=1/2.

## Conclusion in a graph



## Theorem - conclusion

- When P=1 or P=0:

$$\lim_{x\to 0} H=\lim_{x\to 0}-x\cdot\log_2 x-1\cdot\log_2 1=$$

$$\lim_{x\to 0}-\log_2(x^x)=-\log_2 1=0$$

- When P=1/2:

$$H=-\frac{1}{2}\log_2\frac{1}{2}-\frac{1}{2}\cdot\log_2\frac{1}{2}=1$$

## Error in Arithmetic coding

- One bit in the compressed file was changed from "1" to "0":

***In the beginning God created the heaven and the ea*** eithen e
tnos hheaheedh rd ehtesthhfo nh dtohrnh we
telw .e as io
oht o dwf esttfde evsieoeGehee mn n to ito en,nnte
denntoe asipa c., meeha odftdetredd gnt nGp og rmoa oei
ndf r pnfwh eias peedd kaeoweeihd.gomhi G ro m sarc t
eeioorAhiytgawGtLmG r
ddis.i tswot suea df.i. at
i vsha nieI,yawedd D cn eAotc ndeenogidihhese
n etdoleated oiAhettS tDeot Aee retd gctdehneds hheda tdai
sdwL toheAc ohia nrv,iaicna:dde iettweah dea Gmg k.d,ai
kafe rew:ehtd .truit h.ihnd enif e.dwgt hrihrm atn
hSetbhhdd hdtongehie

## Wrong probabilities

- Excess of Huffman over Arithmetic in percents when assuming the rows to compress the columns

| | English | Gadsby | German | Finnish | French |
|---|---|---|---|---|---|
| English | 0.6 | 1.4 | -2.1 | -5.1 | 0.4 |
| Gadsby | -2.4 | 1.0 | -2.9 | -0.8 | -3.3 |
| German | -1.8 | -3.6 | 0.9 | -5.2 | -0.1 |
| Finnish | 2.4 | 1.2 | 2.7 | 0.8 | 2.9 |
| French | 0.8 | 2.2 | -4.6 | -11.3 | 0.9 |