notゲート
・1入力1出力のゲート
・入力が0なら1，1なら0
・出力は0と1
　・出力は1ビット

入力 | 出力
 1 | 0
 0 | 1

・エントロピーは-1/2 log2(1/2) - 1/2 log2(1/2) = 1
・出力はこれ以上圧縮できない

andゲート
・2入力1出力のゲート
・入力が1と1なら1，それ以外なら0
・出力は0と1
　・出力は1ビット

入力 | 出力
 1 1 | 1
 1 0 | 0
 0 1 | 0
 0 0 | 0

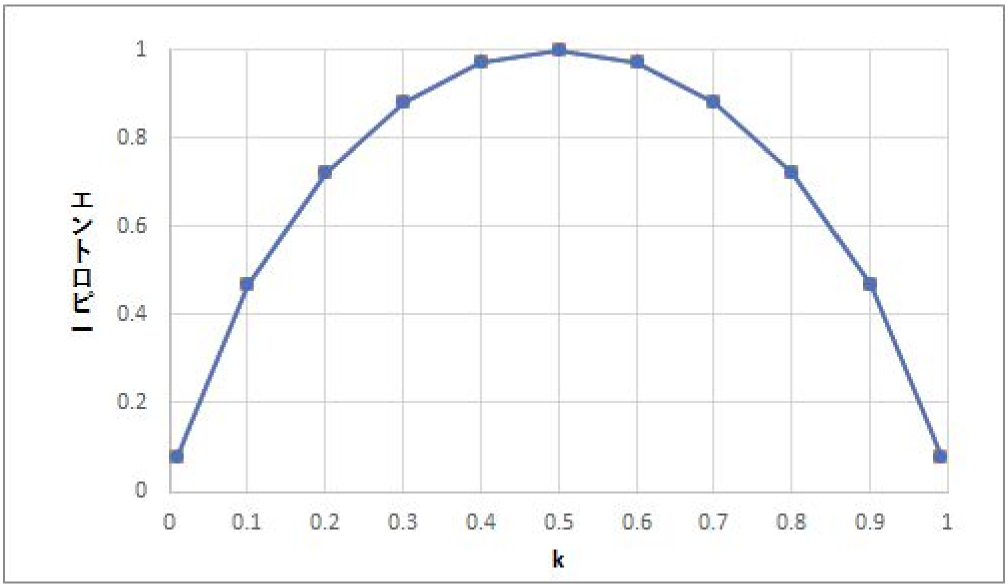
・エントロピーは，-1/4 log2(1/4) - 3/4 log2(3/4) =
1/2 - (3/4 (log2(3) - log2(4))) =
1/2 - 3/4 log2(3) + 3/2 =
2 - 3/4 log2(3) =
0.81
・エントロピーが1ビット未満
　・複数のandゲートがあれば出力を圧縮できる

orゲート
・2入力1出力のゲート
・入力が0と0なら0，それ以外なら1
・出力は0と1
　・出力は1ビット

入力 | 出力
 1 1 | 1
 1 0 | 1
 0 1 | 1
 0 0 | 0
・エントロピーはandゲートと同じ
　・複数のorゲートがあれば出力を圧縮できる

if文

・条件式の真理値によって分岐を行う文
・条件式が真ならthen節，偽ならelse節に分岐する
・の確率をk(0<=k<=1)とした場合，もう一方の確率は1-kで表せる.
・エントロピーは-klog2 k - (1-k) log2 (1-k)となる.



・どちらかの確率が大きくなるにつれてエントロピーは小さくなる

ループの制御情報

ループの制御情報はループが続いたか，ループが続かなかったかの2つであるため，1ビットで表現できる．ループでn回の繰り返しが行われた場合，ループが続いたという情報がn個，ループが続かなかったという情報が1個となり，制御情報全体のビット数はn+1ビットとなる.

ループでn回の繰り返しが行われた場合の制御情報を算術符号を用いて圧縮する．ループが続いたという情報はn個，ループが続かなかったという情報は1個なので，それぞれ確率はn/(n+1)と1/(n+1)となる.

算術符号の平均符号長はエントロピーの値と同等になる．したがって，平均符号長は-(n/n+1)log2(n/n+1) - (1/n+1)*log2(1/n+1)となる.

この平均符号長に元々の制御情報の個数を掛けると，算術符号で圧縮した制御情報のビット数になる．したがって，圧縮後のビット数は-n log2(n/n+1) - log2(1/n+1)のシーリングとなる.

ここでnが十分に大きな数であると，-n log2(n/n+1)はlog2(e)になる．したがって，nが十分に大きな数である場合のビット数はlog2(e) + log2(n)のシーリングとなる.

ループの制御情報は繰り返し回数を用いても表現できる．繰り返し回数を制御情報とした場合，繰り返した回数をnとすると，制御情報の値はn+1となる．n+1を保存するために必要なビットはlog2(n+1)のシーリングである.

制御情報の圧縮に算術符号を用いた場合と繰り返し回数を制御情報に用いた場合では，繰り返し回数を制御情報に用いる方がメモリ効率が良くなる.

$-n \log_2(n/n+1) = -n \log_2(1 - 1/n)$の極限

$\lim(n\to\infty)\ -(1/\log_e 2 * \log(1 - 1/n)) / (1/n) =$ ロピタルの定理による分子分母の微分
$\lim(n\to\infty)\ -(1/n^2)/((1 - 1/n)*\log_e(2)) / (-1/n^2) =$
$\lim(n\to\infty)\ 1 / (1 - 1/n)*\log_e(2) =$
$1 / \log_e(2) =$
$\log_2(e)$