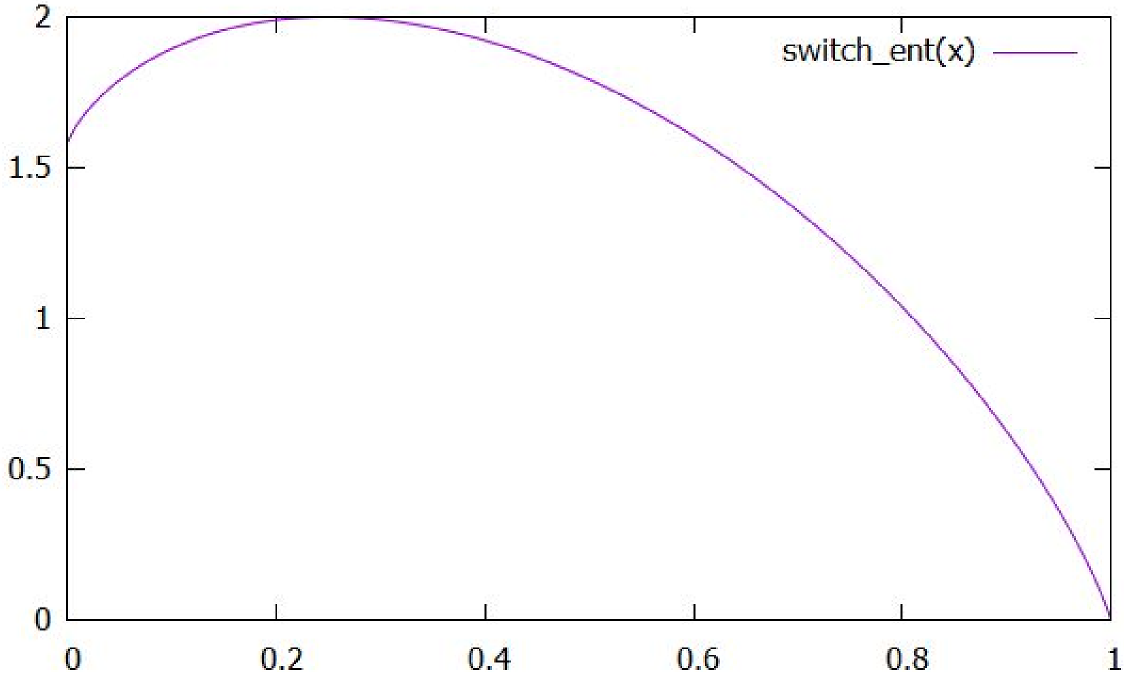
10/27

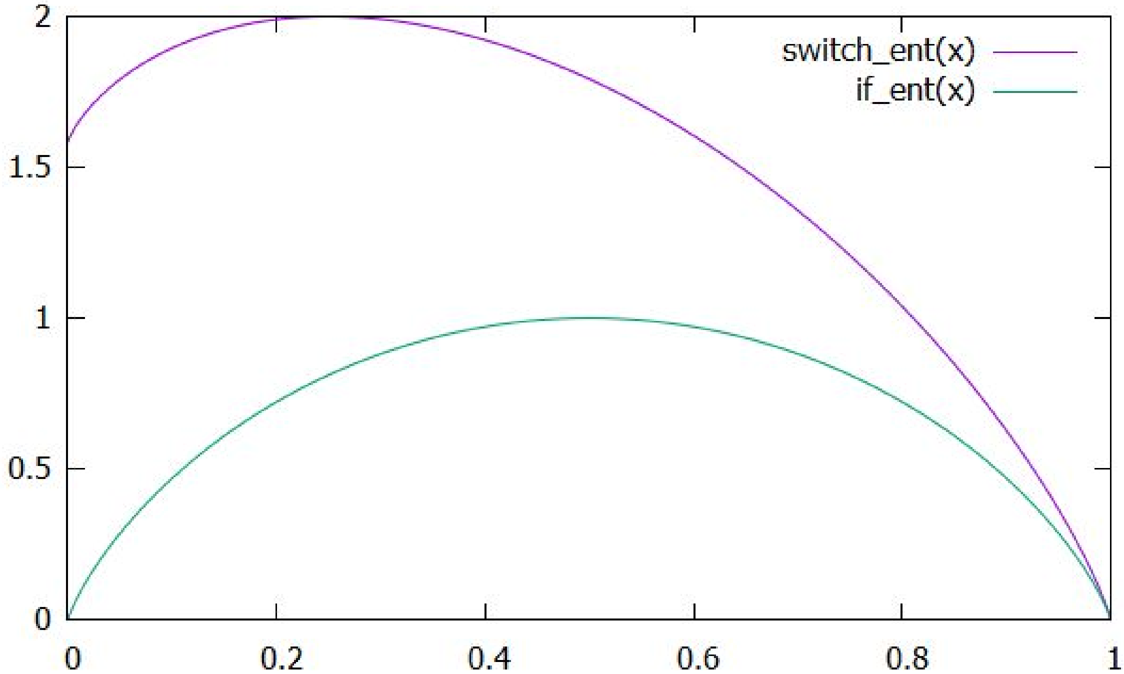
if文にネストされたif文
・どの副文が実行されたかという制御情報は2ビットで表せる
・ネストされたif文がある副文が実行される確率をxとすると，もう一方の副文が実行される確率は1-xで表せる
・ネストされたif文の制御式の値が0とならない確率をyとすると，0となる確率は1-yで表せる
・ネストされたif文が実行されなければ，ネストされたif文の副文は実行されないので，ネストされたif文は条件付きエントロピーとなる
　・if文の副文の実行の確率と，ネストされたif文の実行の確率は互いに独立でない
・各条件付き確率は
$p(y|x) = y$
$p(1-y|x) = 1-y$
$p(y|1-x) = 0$
$p(1-y|1-x) = 0$
となる
・全体のエントロピーは$-x \log_2 x - (1-x)\log_2(1-x) - x(y \log_2 y + (1-y)\log_2(1-y))$となる
・xの値が1に近くても，yの値が0.5に近ければエントロピーは2に近くなる
・xの値が0に近い場合は，ネストされたif文のエントロピーも0に近くなる

switch文
・4つのラベルがあるswitch文を考える
・各ラベルの文が実行される確率をそれぞれa, b, c, dとする
・確率を1つ固定した場合
　・最もエントロピーが高くなるのは残りのすべての確率が同じ場合
　・最もエントロピーが低くなるのは2つの確率が0となる場合
　　・if文のエントロピーと同じ
　　　・p(a)が0なら$-p(a) \log p(a)$も0になるから

・aの値を変えた場合かつエントロピーが最大値となる場合のグラフ



・エントロピーがifと同じになる場合と最大となる場合のグラフ



圧縮できる構成例

・入力の確率分布は一様分布

・andゲート8つ

　・出力の値の確率が互いに独立

・出力は8ビットで表せる
・エントロピーは|~6.48~| = 7ビット
　・圧縮可能

・算術符号で圧縮する場合
　・符号の最小区間幅の値を表現できるだけの精度が必要
　・最小区間幅は最後の出力が1の場合の区間幅
　　・$(3/4)^6 * 1/4$ の中で比率が3/4~1の区間
　・固定小数点数を用いて，どれだけの精度が必要なのか計算すると

$(3/4)^6 * 1/4^2 < 2^x < (3/4)^6 * 1/4$

$\log_2 3^6 - \log_2 4^8 < x < \log_2 3^6 - \log_2 4^7$

$9.5 - 16 < x < 9.5 - 14$

$-6.5 < x < -4.5$

　・この精度は符号部0ビット，整数部1ビット，小数部6ビットの固定小数点数で達成可能

10/22

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
　・入力が一様に分布しているなら
　・入力が一様分布でない場合の考察もする

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ビットあれば表現可能

入力	出力
1 1	1
1 0	0
0 1	0
0 0	0

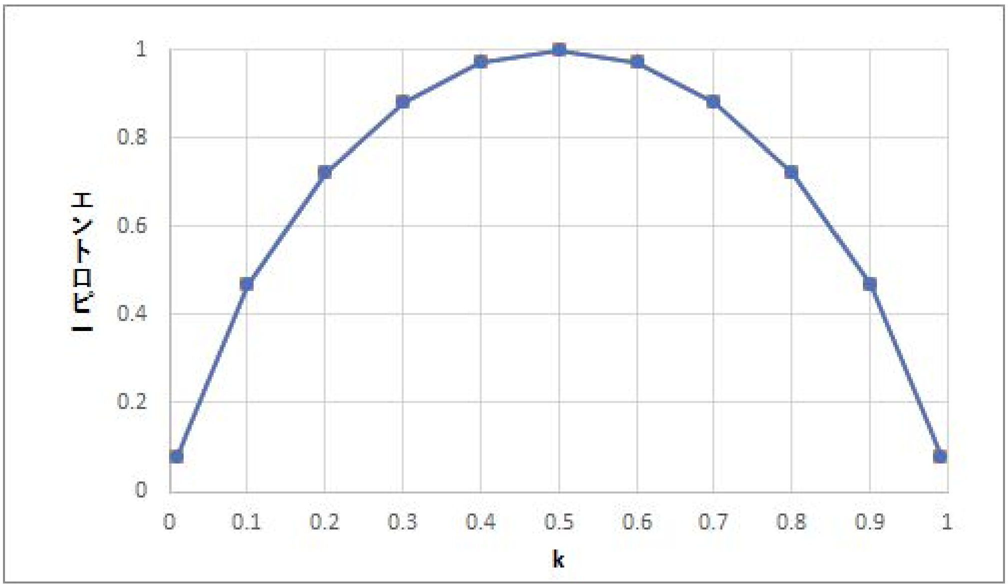
・エントロピーは，-1/4 log2(1/4) - 3/4 log2(3/4) =
1/2 - (3/4 (log2(3) - log2(4))) =
1/2 - 3/4 log2(3) + 3/2 =
2 - 3/4 log2(3) =
0.81
・エントロピーが1ビット未満
　・複数のandゲートがあれば出力を圧縮できる

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ビットで表現可能

入力	出力
1 1	1
1 0	1

０１｜　1
００｜　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
・真の確率を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) ~|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.

$-n \log_2(n/(n+1)) = \log_2(1+1/n)^n \to \log_2 e$
$n \to \infty$

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

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

$(\log x)' = 1/x$
$(g(f(x)))' = g'(f(x)) * f'(x)$
$(\log_e f(x))' = 1/f(x) * f'(x)$

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

$\lim(n\to\infty)$ $1 / (1 - 1/(n+1))*\log_e(2) =$
$1 / \log_e(2) =$
$\log_2(e)$