情報理論11

藤田 一寿

津山工業高等専門学校情報工学科 講師
電気通信大学先進理工学科 協力研究員

ver.20170705

雑音のない離散通信路の符号化定理

通信路の容量と信号

- 互いに独立に都合のよい確率で信号が発せられた場合，通信路は能力をいっぱいに使っての情報を送れる．
  - そんなに都合よく信号は発生しない．
- 通信路の能力を限界まで使える良い信号に変換する方法(符号化)を考えればよいのではないか．

雑音のない通信路の基本定理

1文字当たりHのエントロピーを持つ情報源と単位時間あたりCの容量を持つ通信路があるとき,この通信路を用いて単位時間あたり$(C/H-\varepsilon)$文字の割合で情報を送れるような符号化がどんな小さな$\varepsilon>0$に対しても存在する.

理想的な符号化が必ず存在する.

符号化による冗長度の除去

英語における1文字当たりの情報

- 英語のアルファベット1文字あたりの情報(それぞれの文字が独立に等確率に生成される場合)
  - 4.7ビット
- 実際の英語におけるアルファベット1文字あたりの情報
  - 1.3ビット程度
- あまり使われない文字がある,文字の並びに規則性がある,などの理由で情報量が小さくなる.

符号化

- 実際の情報源は無駄が多い(冗長度が大きい)
- 通信速度を上げるためには冗長度を減らしたムダのない表現が必要
- 符号化をし冗長度を減らす.
  - 符号化とは情報源から得られた文字列を別の文字列に置き換えること.

冗長度の削減

例として，A，B，C，Dの４つの文字を発生させる情報源を考える．各文字が発生する確率pA, pB, pC, pDを

$$p_A = \frac{1}{2}, p_B = \frac{1}{4},$$
$$p_C = \frac{1}{8}, p_D = \frac{1}{8},$$

とする．この情報源の冗長度の削減を行う．

この情報源の一文字あたりのエントロピーは

$$\begin{aligned} H &= -\frac{1}{2}\log\frac{1}{2} - \frac{1}{4}\log\frac{1}{4} - \frac{1}{8}\log\frac{1}{8} - \frac{1}{8}\log\frac{1}{8} \\ &= \frac{7}{4} \end{aligned}$$

もし,各文字が独立に等確率で発生するのであればそのエントロピーH0は

$$H_0 = -\frac{1}{4}\log\frac{1}{4} \times 4 = \log 4 = 2$$

となる.これはこの情報源が持てる最大のエントロピーである.

これらから,この情報源の冗長度rは

$$r = 1 - \frac{H}{H_0} = \frac{1}{8}$$

である.よって,うまく符号化すれば同じ情報を表すのに文字列を1/8だけは短くすることが出来る.

2進符号化

情報源から発生する文字系列を０と１の２つの信号からなる系列に変換する．例えば次のような符号化を考える．

文字	符号語
A	$\to 0$
B	$\to 10$
C	$\to 110$
D	$\to 111$

文字列を2進に変換する符号化を2進符号化という. そして,このような長さが一定でない符号を可変長符号という.

AABCDABA

という先の変換則にもとづいて文字列を符号化してみる

0 0 10 110 111 0 10 0

14文字の系列が得られる.

情報源は1文字当たり7/4ビットの情報量を持っているので,8文字の文字列全体で7/4X8=14ビットの情報を持っている.

14文字の2進符号は14ビットの情報を持っているので完全に冗長性は排除できている.

別のルールで復号化してみると

AABCDABA

A: 0

B: 10

C: 110

D: 111

符号化

00101101110100

復号化

00:A

01:B

10:C

11:D

ACDBDCA

7文字に減り冗長度が減ったことが分かる.

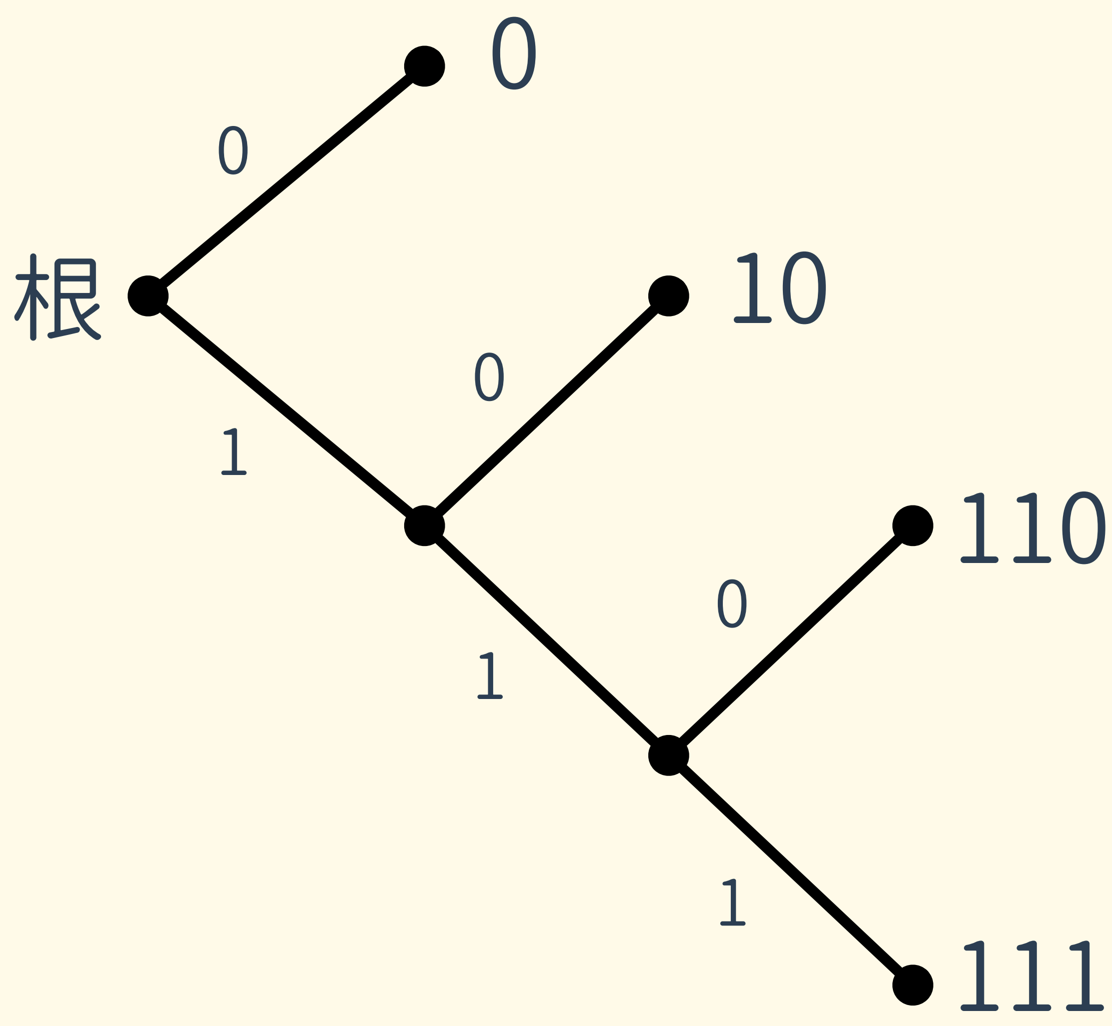
減ったことを確認するため，あえて別の文字に復号化してみた

符号木

- 元の文字列に戻せる(復号可能な)符号化をしなければならない.
- 符号木を使えば復号可能な符号を作ることができる.
  - 同じ符号が異なった文字系列に割り当てられていない.
  - 切れ目がわからなくなって解読不能にならない.

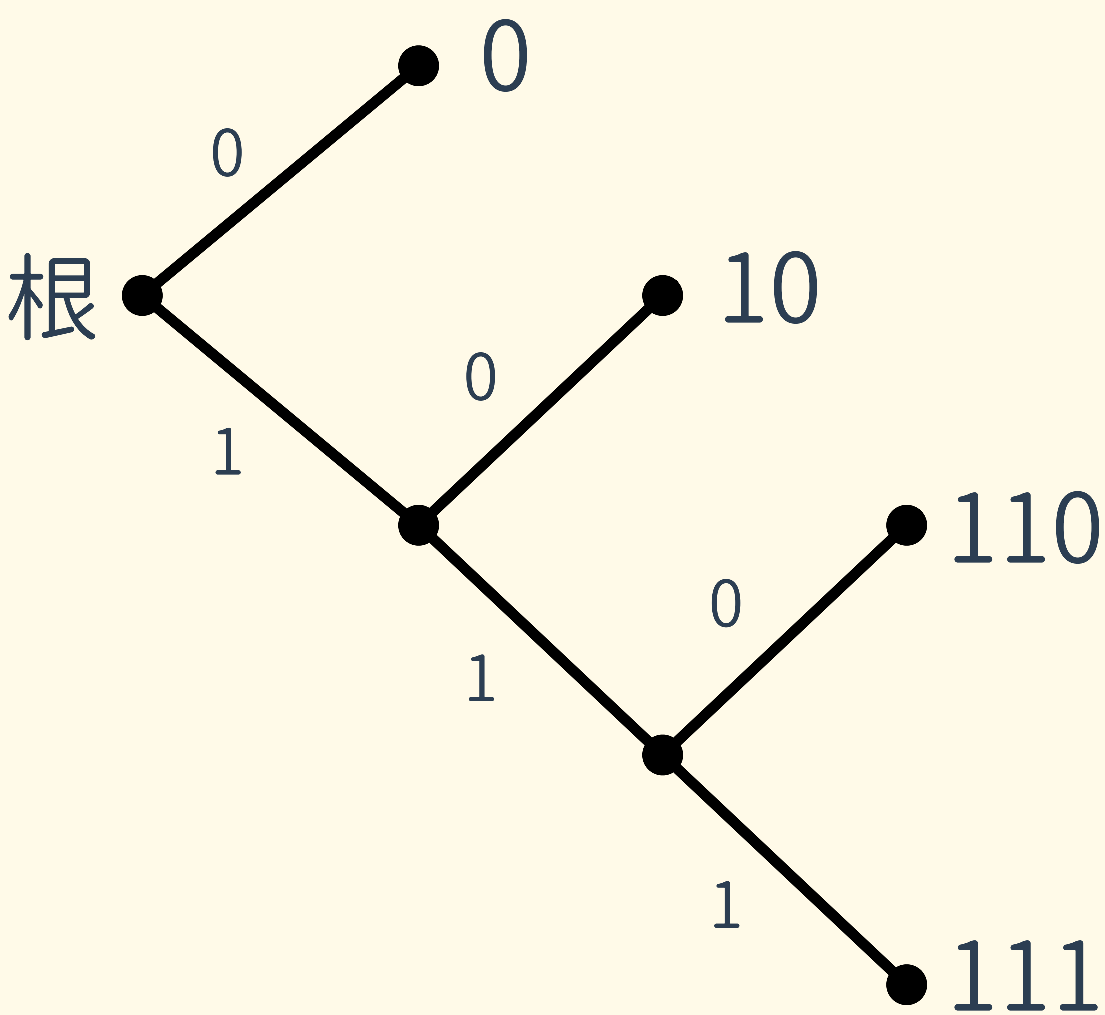
符号木

- 符号木
  - 接点:●
  - 枝:線
  - 葉:末端

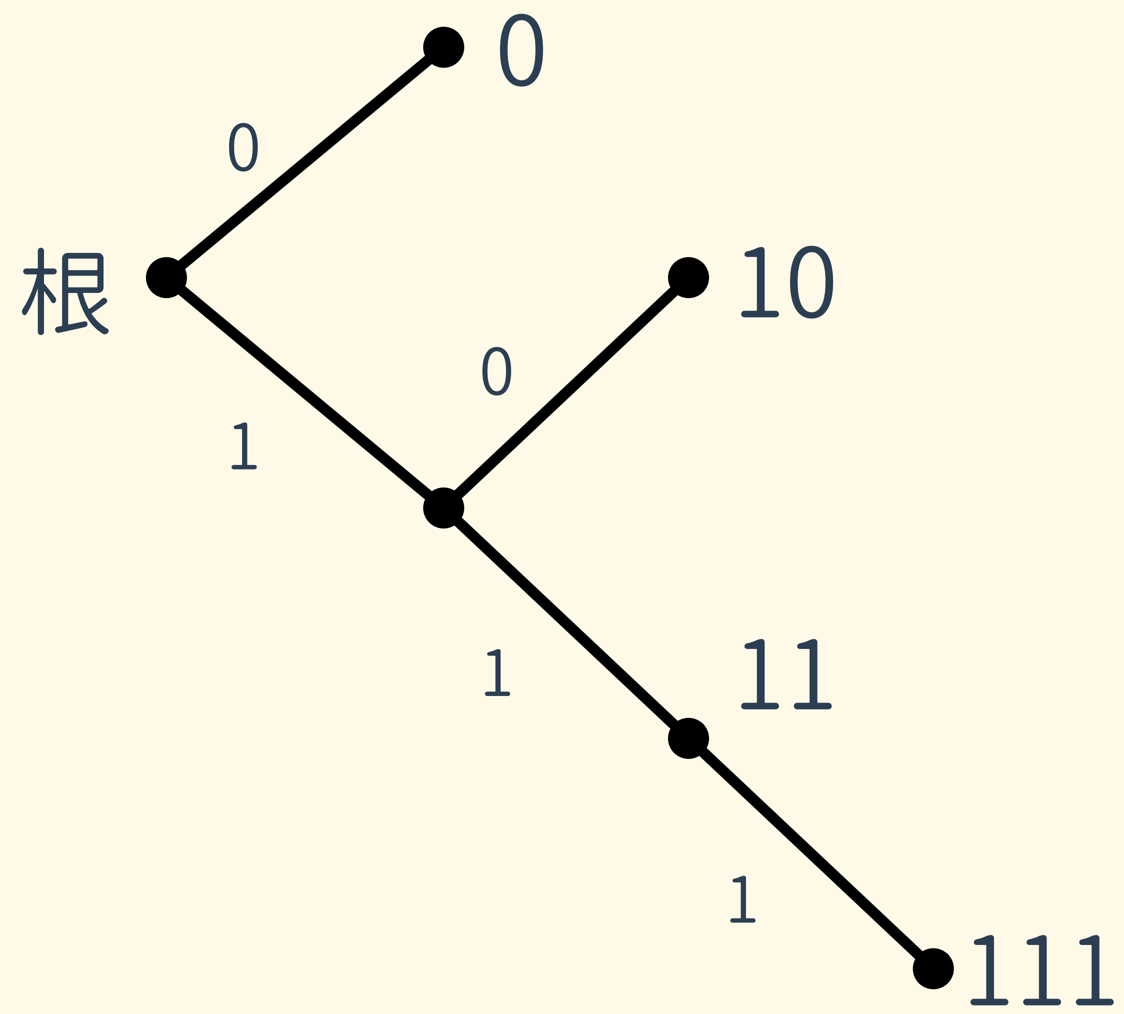


- 木の根から0,1の分岐にしたがって順にたどって葉に行き当たったら文字に置き換え,また根からたどるという操作する.符号語が葉のみからなる符号を語頭符号と呼ぶ.

符号木



復号可能な場合



復号不可能な場合

クラフトの不等式

長さl1, l2,⋯, lmとなるM個の符号語を持つr元符号が一意復号可能なら

$$\sum_{i=0}^{M} r^{-l_i} \leq 1$$

が成り立つ.

先ほどの符号化の例で確かめる

符号長はそれぞれ、1，2，3，3である。

符号は2元符号である。

以上から

$$2^{-1} + 2^{-2} + 2^{-3} \times 2 = 1$$

よってクラフトの不等式を満たす。

文字　符号語

$$A \to 0$$

$$B \to 10$$

$$C \to 110$$

$$D \to 111$$

- 一意複合可能
- 瞬時符号

シャノン符号化

- 情報源から出てくる長さNの文字系列がある．
- 文字の種類はk個であるとする．
- 各文字独立に同じ確率で生じたとした場合のエントロピーH0は

$$H_0 = \log k$$

- 文字列のエントロピーは文字系列の長さがNなので

$$NH_0 = N\log k$$

- よって文字系列の総数は

$$2^{H_0 N} = k^N$$

$$2^{H_0 N}$$

- ただ実際には，すべての文字が均等に出るとは限らない．
- よく出てくる文字系列の数は 2^{HN} 個でそれ以外はめったに出ない(大数の法則)．
- よく出てくる文字系列を2進符号で表せば効率が良いのではないか．
- HN桁の2進符号で表せば無駄なく大抵の文字系列を表せる．
- 一文字当たりH桁の2進符号

シャノン符号化のやり方

1. 出現確率の高い順に文字を並べ,これを改めてi=1, 2, …, kで表す.

$$P_1 \geq P_2 \geq \cdots \geq P_k$$

2. 文字iまでの累積確率を計算する.

$$F_i = \sum_{j=1}^{i-1} P_j$$

3. Fiの2進数表示の小数点以下 $l_i = \lceil \log(1/P_i) \rceil$ ビットを文字の符号とする.

$\lceil x \rceil$ xを以上の最小の整数

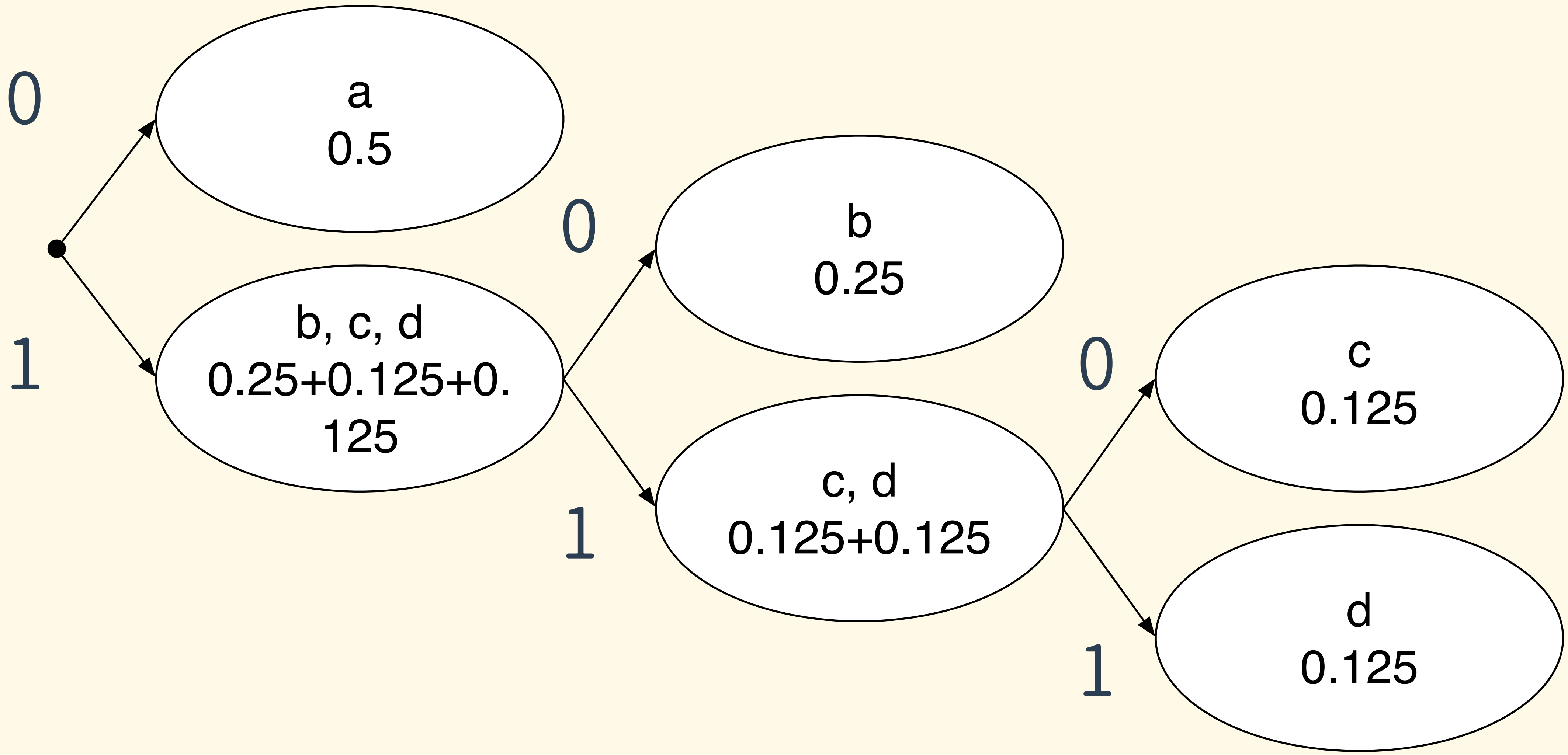
文字i	確率Pi	符号長li $l_i = \lceil \log(1/P_i) \rceil$	累積確率Fi	2進数	符号語
a	0.55	1	0	0.0	0
b	0.25	2	0.55	0.10001	10
c	0.15	3	0.8	0.11001	110
d	0.05	5	0.95	0.11110	11110

ファノ符号化

1. 文字の出現確率の和がほぼ1/2になるように文字を2グループに分ける.
2. それぞれの中で、出現確率の和がなるべく等しくなるように文字を2グループに分ける.
3. 分けられた全てのグループの中の文字が1文字だけのなる状態までステップ2を繰り返す.
4. 分割の順番に従い木を作り符号を作る.

ファノ符号化の例

アルファベット{a, b, c, d}の各文字の出現確率が{0.5, 0.25, 0.125, 0.125}である情報源をファノ符号化する.



結果a→0, b→10, c→110, d→111という符号が得られる.

最適な符号化

- 2つの文字i, jの出現確率をPi，Pj，符号語の長さをli，ljとする。Pi>Pjの場合、li<=ljである．
- 出現確率が1番目と2番目に低い符号語の長さは等しい．
- 出現確率が1番目と2番目に低い符号語は最後の文字だけ異なる．

ハフマン符号化

- 出現率の順に文字を並べる.
- 確率の低い順に2つの文字を選び0,1を割り当てる. この2つの文字を統合して新しい文字を作る. その出現確率は2つの文字の確率の和として新しいアルファベットを作る.
- アルファベットの数が2つになるまでステップ2を繰り返す.
- 統合するときに割り当てられた0, 1を結合して符号を作成する.

ハフマン符号化の例