

1.3 瞬時復号可能な符号

瞬時復号可能な符号とは、名前の通り、符号がやってきたらそれを瞬時に復号可能な符号のことをいう。これはつまり、全てのメッセージを受信してから復号を行うのではなく、来た時点で遅延なく復号可能な符号のことをいうのである。厳密に言えば、符号 C が瞬時復号可能な符号 (instantaneously decodable code) あるいは瞬時符号 (instantaneous code) であるとは、各符号列 $w_{i_1}, w_{i_2}, \dots, w_{i_n}$ に対し、 $t = w_{i_1} w_{i_2} \dots w_{i_n} \dots$ で始まる全ての符号列が $s = s_{i_1} s_{i_2} \dots s_{i_n} \dots$ として一意に復号され、その後続く t のシンボル列に無関係なことをいう。

例 1. 2 元符号 C

$$s_1 \mapsto 0, s_2 \mapsto 10, s_3 \mapsto 11$$

は瞬時復号可能な符号、つまり、メッセージ t を受信しながら復号できる。なぜならば、0 が来れば s_1 に復号可能であるし、1 が来れば、次に来る符号を見てすぐに s_2 なのか s_3 なのかを判定できる。

例 2. 2 元符号 C

$$s_1 \mapsto 0, s_2 \mapsto 01, s_3 \mapsto 11$$

は瞬時復号可能ではない。たとえば、 $t = 01111\dots$ というメッセージが来たとき、 $t = 0.11.11.11\dots$ なのか $t = 01.11.11.11\dots$ なのかはすぐには判定できず、メッセージ長が奇数か偶数かを見なければ判定することはできない。

一般に、瞬時符号であれば一意復号可能な符号であることは成立するが、逆は成立しない。例えば、例 1 で定義する符号は一意復号可能だが、既に見たように瞬時復号可能ではない。

符号 C が語頭符号 (prefix code) であるとは、どの符号語 w_i も他の符号 w_j ($i \neq j$) の語頭にないことをいう。すなわち

$$\forall w \in T^*, w_j \neq w_i w$$

である。また、 $C_1 = \emptyset$ とも言い換えることもできる。この語頭符号と瞬時符号との間には次のような関係がある。

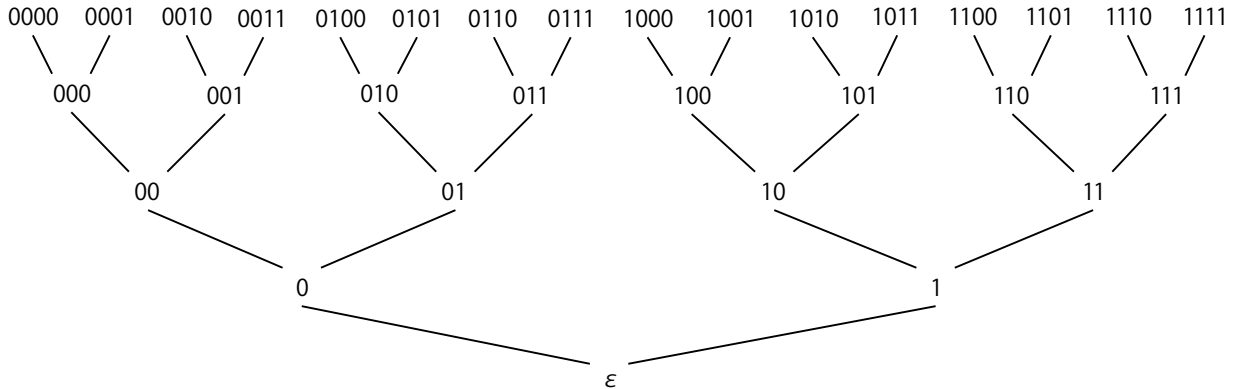


図 1 語長 4 のときの 2 分木. 必ず根に近いノードの方をそのノード以降のノードは語頭として持っている.

定理 1. 符号 C が瞬時符号であることと, 語頭符号であることは同値.

Proof. まず, C が瞬時符号であるとして, 語頭符号となることを示す. C が語頭符号でないとする. このとき, ある w_j の語頭に w_i という符号列が含まれる. すると, $t = w_i \cdots$ で始まる符号列は $s = s_i \cdots$ にも, $s = s_j \cdots$ にも復号化される. よって, 符号 C は瞬時符号ではないが, C が瞬時符号であることに矛盾. 以上から, 符号 C は瞬時符号であれば, 語頭符号となる.

次に, C が語頭符号であれば, 瞬時符号でもあることを確認する. C が語頭符号なので, $t = w_i \cdots$ であれば, この符号列は $s = s_i \cdots$ に復号化されなければならない. これは w_i は他の w_j ($i \neq j$) の語頭とはならないために, 一意に復号化されるためである. よって, どの符号列に対してもこの事実が成立するので, 全てのシンボル列は t の後続のシンボル列とは無関係となる. 以上から, C は瞬時符号となる. \square

1.4 瞬時符号の構成法

r 元瞬時符号は r 分木によって構成することができる. ここで考える r 分木は図 1 のように根に空語 ϵ を配置し, それに各シンボルを 1 つだけ加えたものを根の子にそれぞれ追加し, 以下同様にシンボルを 1 つずつ加えたものをそれぞれ子に... ということを繰り返す.

例 3. 語長を 4 までとしたときに構成される 2 分木は図 1 の通り.

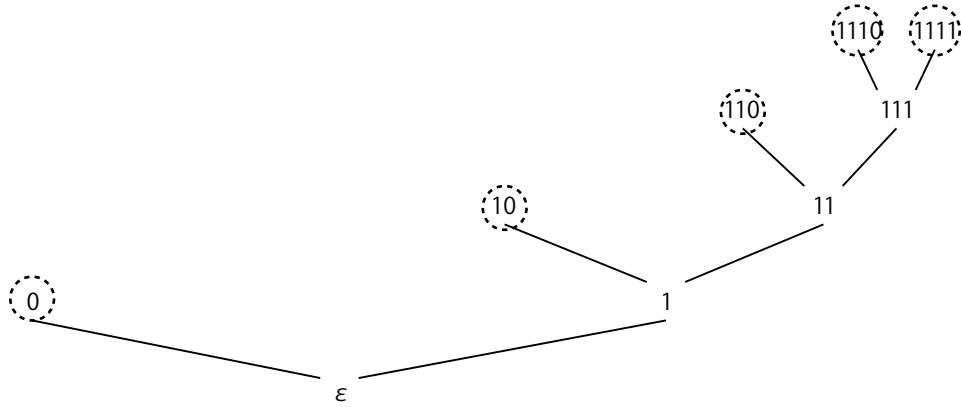


図 2 例 3 からいくつか刈り込んで瞬時符号を構成している例. 符号として選ばれたノードは円で囲まれている.

このようにして作った r 分木に対して, 先程の語頭符号と瞬時符号が同値である命題を使い, 望む瞬時符号を構成していく. まず, r 分木のノードを 1 つ選ぶ. 次に, 選んだノード以降のノードを刈り込む (選んだノードの子や孫を削除する). すなわち, 選んだノードを語頭に含んでいるノードを全て木から削除する. これを繰り返すことで, 最終的に瞬時符号を構成することができる.

例 4. 例 3 の中から 0, 10, 110, 1110, 1111 を符号語とした符号は瞬時符号となる. 図 2 の通り, 選択したノードが必ず他の語を語頭とならないよう刈り込みがなされている.

1.5 クラフトの不等式

瞬時符号を r 分木から構成するには, 除かれる葉の割合が 1 以下でなければならないという当然の制限が存在する.

クラフトの不等式 (Kraft's inequality)

定理 2. 符号長が l_1, \dots, l_q であるような r 元瞬時符号 \mathcal{C} は, 次の不等式を満たすとき, かつ, そのときのみ存在する.

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1$$

Proof. まず, 不等式を満たす符号が瞬時符号となることを示す. 一般性を失うことなく,

$l_1 \leq l_2 \leq \dots \leq l_q$ と仮定する. また, $l = \max\{l_1, \dots, l_q\}$ とし, T^* の高さ l までの部分を $T^{\leq l} = \bigcup_{i=0}^l T^i$ とする ($T^i = \{w : |w| = i\}$). この集合から先程の r 分木を作る手順で木をつくる. この木の高さ l_1 において, ある符号 w_1 を s_1 に割り当ててることを考える. すると, その w_1 を語頭に持つ高さ $l_1 + 1$ 以上のノード, つまり, w_1 を頂点とした部分木は刈り込まれる. r 分木であることから刈り込まれる部分木の葉は r^{l-l_1} 個ある. ここで, 不等式

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1$$

の両辺に r^l をかけると,

$$\sum_{i=1}^q r^{l-l_i} \leq r^l$$

となる. $q > 1$ に対し, 明らかに

$$\frac{1}{r^{l_1}} < \sum_{i=1}^q \frac{1}{r^{l_i}} \iff r^{l-l_1} < \sum_{i=1}^q r^{l-l_i}$$

となる. 従って,

$$r^{l-l_1} < \sum_{i=1}^q r^{l-l_i} \leq r^l$$

となる. この刈り込みを $k < q$ となる k まで続けると, 刈り込まれた葉の数は $\sum_{i=1}^k r^{l-l_i}$ となるが, これは

$$\sum_{i=1}^k r^{l-l_i} < \sum_{i=1}^q r^{l-l_i} \leq r^l$$

を満たす. この不等式からまだ刈り込まれていない葉が存在することがわかるので, その葉に繋がっている w_{k+1} を選択することができる. これによって, q まで符号語を選択することができる. この過程を通して, 選択された符号語を語頭として持つものは全て刈り込んで次のステップに移っていることから, 選ばれた符号語を持つ符号は語頭符号となる. よって, 不等式を満たす符号語は瞬時符号となることが示された.

次に, 符号 C が瞬時符号であるとして不等式を満たすことを示す. C は瞬時符号なため, 語頭符号となる. 従って, $T^{\leq l}$ の葉は C に属する多くとも 1 つの符号しか語頭に持たない.

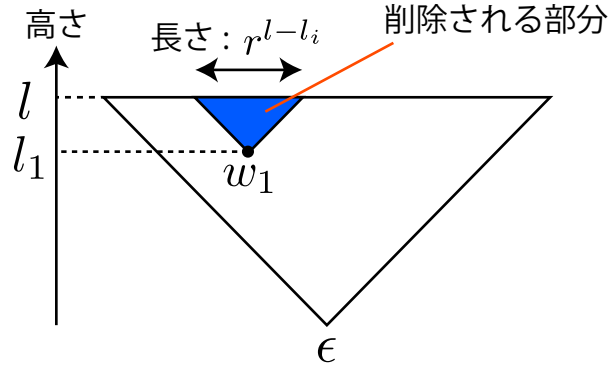


図3 w_1 を頂点とする部分木は r^{l-l_i} 枚の葉を持つ.

\mathcal{C} に属する符号語 w_i は r^{l-l_i} 枚の葉の語頭となるので, $\sum_{i=1}^q r^{l-l_i}$ 枚の葉が \mathcal{C} のいずれかの符号語を語頭に持つことになる. $T^{\leq l}$ は r^l 枚の葉を持つので, 不等式

$$\sum_{i=1}^q r^{l-l_i} \leq r^l$$

が成立する. この不等式の両辺を r^l で割れば示したかった不等式が導出される. \square

1.6 マクミランの不等式

先のマクミランの不等式と同じような不等式が, 実は一意復号可能な符号に対しても成立する.

マクミランの不等式 (McMillan's inequality)

定理 3. 符号長が l_1, \dots, l_q な一意復号可能な r 元符号 \mathcal{C} は不等式:

$$\sum_{i=1}^q \frac{1}{r^{l_i}} \leq 1$$

を満たすとき, かつ, そのときのみ存在する.

Proof. 不等式を満たすときには瞬時符号が存在するが, 瞬時符号は一意復号可能な符号となるため, 一意復号可能な符号が存在する場合に, 不等式が成立することを示す.

まず,

$$\bullet K = \sum_{i=1}^q \frac{1}{r^{l_i}}$$

- $L = \max\{l_1, \dots, l_q\}$
- $l = \min\{l_1, \dots, l_q\}$

とする. ここで, K^n を考える. この K^n は $\prod_{j=1}^n r^{-l_{i_j}}$ という項の和で表される. さらに,

$$j = \sum_{j=1}^n l_{i_j}$$

とする. 全ての i に対し, $l \leq l_i \leq L$ なので, $ln \leq j \leq Ln$ となるので, $N_{j,n}$ を j の重複を許した表現の仕方の数とすれば,

$$K^n = \sum_{j=ln}^{Ln} \frac{N_{j,n}}{r^j}$$

と表すことができる. $N_{j,n}$ は n 個の符号列 w_{i_1}, \dots, w_{i_n} を長さが j となるよう列の個数と同じとなる. その列の各々は長さが j となる符号列 $t = w_{i_1} \dots w_{i_n}$ を決定するが, 符号 \mathcal{C} は一意復号可能であるため, 各 t の表現の仕方は 1 通りしかない. したがって, $N_{j,n}$ は t の個数に等しいか少なくなる. よって, $N_{j,n} \leq r^j$ となる. K^n は $Ln - ln + 1$ 個の項 $N_{j,n}/r^j \leq 1$ の和なので, 任意の $n \geq 1$ に対し,

$$K^n \leq Ln - ln + 1 = (L - l)n + 1$$

となる. K, l, L は n とは独立なため, $K > 1$ ならば, K^n は指数関数的に増大する. しかし, n に線形な式で抑えられているため, 指数関数的に増大することはない. よって, $K \leq 1$ でなければならない. 従って, K の定義を思い出せば, 望んでいた不等式を得ることができる. \square

このマクミランの不等式とクラフトの不等式から, 次の系がすぐに導ける.

系 1. 符号長が l_1, \dots, l_q の r 元瞬時符号は, これらの符号長をもつ, 一意復号可能な r 元符号が存在するとき, かつ, そのときのみ存在する.

この系自体は同じ不等式を, 一意復号可能な符号の存在も瞬時復号可能な符号の存在も, 同じ不等式を満たさなければならず, また, 存在すれば, 同じ不等式が満たされることからすぐに従うことがわかる.