

1 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。

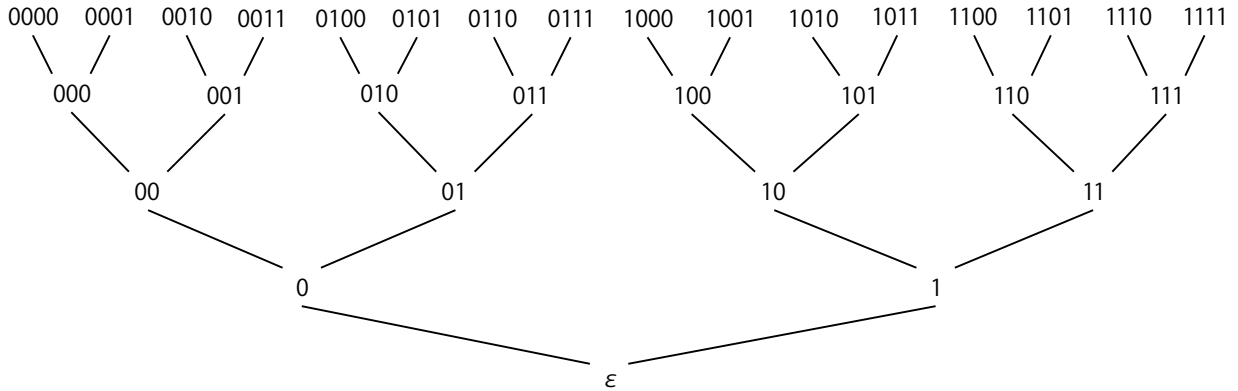


図 1 語長 4 のときの 2 分木. 必ず根に近いノードの方をそのノード以降のノードは語頭として持っている.

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

2 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 ?? のようにより根に空語 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.

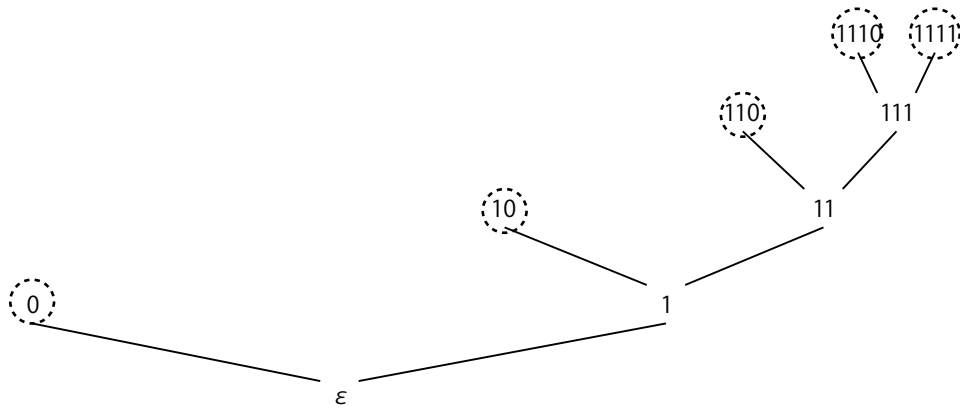


図 2 例 3 からいくつか刈り込んで瞬時符号を構成している例. 符号として選ばれたノードは円で囲まれている.

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.

3 クラフトの不等式