

## 本講義の概要

- 情報通信の数理

- ★ 情報理論

- ★ 符号理論

- ★ 暗号理論

- それを支える数学

- ★ 有限体とその上の線型代数・Galois 理論

- ★ 情報量の理論・計算量の理論

## 情報通信を行なう際の要請

- 効率的に → 情報理論
- 確実に → 符号理論
- 安全に → 暗号理論

## 情報理論 (情報源符号化・情報量の理論)

- 伝えるべき情報をより効率良く伝えるには
- 「効率の良さ」を計る
  - ★ 伝えるべき「情報の量」を計る
  - ★ 伝える為の「手間」を計る

→ **Shannon**

「情報の量は伝えるのに必要な手間と一致」

## 情報の符号化

アナログデータ (連続データ)

↓ サンプルングなど

デジタルデータ (離散・有限データ)

↑ ここで扱う “情報” (元データ・情報源)

↓ ← ここで扱う “符号化”

通信用の形式のデジタルデータ

(特定 (一般には少数) の種類の文字の列)

## 文字データ → **ASCII code**

	0	1	2	3	4	5	6	7	8	9	a	b	c	d	e	f
0	(control characters)															
1	(control characters)															
2		!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
3	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6	'	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7	p	q	r	s	t	u	v	w	x	y	z	{		}	~	

**7 bit** で 1 文字を表す (実装上は **8 bit** にする)

## ASCII code

<b>A</b>	01000001	<b>J</b>	01001010	<b>S</b>	01010011
<b>B</b>	01000010	<b>K</b>	01001011	<b>T</b>	01010100
<b>C</b>	01000011	<b>L</b>	01001100	<b>U</b>	01010101
<b>D</b>	01000100	<b>M</b>	01001101	<b>V</b>	01010110
<b>E</b>	01000101	<b>N</b>	01001110	<b>W</b>	01010111
<b>F</b>	01000110	<b>O</b>	01001111	<b>X</b>	01011000
<b>G</b>	01000111	<b>P</b>	01010000	<b>Y</b>	01011001
<b>H</b>	01001000	<b>Q</b>	01010001	<b>Z</b>	01011010
<b>I</b>	01001001	<b>R</b>	01010010		

## モールス符号 (Morse code)

A	· -
B	- ...
C	- · - ·
D	- ..
E	·
F	.. - ·
G	- - ·
H	....
I	..

J	· - - -
K	- · -
L	· - ..
M	- -
N	- ·
O	- - -
P	· - - ·
Q	- - · -
R	· - ·

S	...
T	-
U	.. -
V	... -
W	· - -
X	- .. -
Y	- · - -
Z	- - ..

## モールス符号 (Morse code)

### 1 文字のための符号長が区々

← 頻度の高い文字は短く、低い文字は長く

→ 頻度まで考慮して符号長の期待値を短く

… 頻度 (出現確率) を考慮して  
符号効率の定式化を考えている



## モールス符号 (Morse code)

### 1 文字のための符号長が区々

→ 1 文字毎の区切りは判るのか？

→ 実はモールス符号では、  
文字間・単語間の送信間隔が  
定められている

A	· -	H	....	O	- - -	U	.. -
B	- ...	I	..	P	· - - ·	V	... -
C	- · - ·	J	· - - -	Q	- - · -	W	· - -
D	- ..	K	- · -	R	· - ·	X	- .. -
E	·	L	· - ..	S	...	Y	- · - -
F	.. - ·	M	- -	T	-	Z	- - ..
G	- - ·	N	- ·				

例: the cat and the dog

-     · · · ·     ·                   - · - ·     · - - -  
· -     - ·     - ..                   -     · · · ·     ·                   - ..  
- - -     - - ·   ( は · 1 個分の間隔)

→ - と · と   とを用いて符号化している

## 情報源符号化の定式化

情報源 **alphabet**  $S$  : 有限集合

$S^+ := \bigsqcup_{n \geq 1} S^n$  :  $S$  の元の 1 個以上の列

$S^0 := \{\varepsilon\}$  : 空語

$S^* := \bigsqcup_{n \geq 0} S^n$  :  $S$  の元の 0 個以上の列

$$= S^+ \sqcup \{\varepsilon\}$$

$w \in S^n$  に対し、 $|w| := n$  (文字列の長さ)

## 情報源符号化の定式化

符号 (伝送) **alphabet**  $T$  : 有限集合  
(しばしば  $T = \{0, 1\}$ )

$C : S \longrightarrow T^+ : \text{符号 (code)}$

$C$  の像の元: **符号語 (code word)**

→ 文字列を並べて  $C^* : S^* \longrightarrow T^*$  に延長

## 符号への要請

- 一意復号可能 (uniquely decodable) か？
- 一意復号可能とした上で、  
瞬時復号可能 (instantaneously decodable)  
か？
- その上で効率が良いか？

## 一意復号可能でない例

$$S = \{a, b, c\}, T = \{0, 1\}$$

$$a \longmapsto 0$$

$$b \longmapsto 01$$

$$c \longmapsto 001$$

「001」が **ab** か **c** か判らない

→ 一意復号可能で**ない**!!

## 瞬時復号可能でない例

$$S = \{a, b, c\}, T = \{0, 1\}$$

$$a \mapsto 0$$

$$b \mapsto 01$$

$$c \mapsto 11$$

一意復号可能ではあるが、

「011...」まで見ただけでは

ac... か bc... か判らない

(「0111」なら bc、「01111」なら acc)

→ 瞬時復号可能でない!!

## 瞬時復号可能な例

$$S = \{a, b, c\}, T = \{0, 1\}$$

$$a \mapsto 0$$

$$b \mapsto 10$$

$$c \mapsto 11$$

$$\longrightarrow \mathcal{C}(S) = \{0, 10, 11\} \subset T^+$$

だけを見て判る特徴があるか？



## 符号への要請

- 一意符号:

$$C^* : S^* \longrightarrow T^* : \text{単射}$$

- 瞬時符号:

$$C^*(x) = C(s)w \implies x = sy$$

(最初に届いた符号語  $C(s)$  で  
最初の文字  $s$  が復元できる)

- 効率が良い... 符号長  $|C(s)|$  が小さい

## 瞬時符号の性質

- $C$  : 瞬時符号  $\implies C$  : 一意符号
- $C$  : 瞬時符号  $\iff C$  : 語頭符号  
( $C(s') = C(s)x \implies s' = s, x = \varepsilon$ )

## 瞬時符号の作り方

「符号語木」を考えよう