# チューリングマシン入門と ふぃっしゅ数 Ver.4

岩上　裕哉

神戸大学理学部物理学科 1 年生

2017 年 3 月 4 日

# 発表の流れ

1. はじめに
   - 動機
2. 巨大数概要
   - グラハム数
   - アッカーマン関数
   - FGH、巨大数の比較
3. チューリングマシン
   - チューリングマシン
   - 停止問題
   - 神託
4. ふぃっしゅ数 Ver.4
   - ビジービーバー関数
   - 神託機械 (oracle machine)
   - ふぃっしゅ数 Ver.4

# 動機

- 巨大数論がやりたい！（$F_4$ を定義したい）
- 情報学を学びたい

| | | |
|---|---|---|
| 計算不可能関数 | $F_4$ | $F_7$ |
| 計算可能関数 | $F_5$ | $F_6$ |
| | $F_3$ | |
| | $F_1$ | $F_2$ |

# 巨大数（関数論）とは何か？

巨大数論とは…

- 大きな数を作る過程を楽しむ
- 大きな数を比較して楽しむ

# どの程度の大きさなのか？

観測可能な全宇宙の素粒子数 $10^{80}$ < グラハム数 $G(64)$

## Definition 定義 (（クヌースの）矢印表記)

$$a \uparrow^1 b = a^b$$

$$a \uparrow\uparrow b = \underbrace{a \uparrow a \uparrow \ldots \uparrow a}_{b\ \text{個の}\ a}$$

$$a \uparrow\uparrow\uparrow b = \underbrace{a \uparrow\uparrow a \uparrow\uparrow \ldots \uparrow\uparrow a}_{b\ \text{個の}\ a}$$

$$a \uparrow^n b = a \underbrace{\uparrow\uparrow \ldots \uparrow}_{\text{n 本}} b$$

# グラハム数

## Definition 定義 (グラハム数)

$$G(x) \stackrel{\text{def}}{=} 3 \uparrow^{x} 3$$

$$G^{n}(x) = \underbrace{G(G(G(\dots G(x)\dots)))}_{n \text{ 個の } G(x)}$$

グラハム数 $G^{64}(4)$ 原始再帰関数の典型例

# アッカーマン関数

**Definition 定義 (アッカーマン関数)**

非負整数 $x, y$ に対し、以下のように定義される関数 $A(x, y)$ をアッカーマン関数とする。

$$A(0, y) = y + 1$$

$$A(x + 1, 0) = A(x, 1)$$

$$A(x + 1, y + 1) = A(x, A(x + 1, y))$$

$$A(1, 1) = 3 \qquad A(1, 3) = 5 \qquad A(1, y) = y + 1$$

$$A(2, y) = 2y + 3 \qquad A(2, 1) = 5 \qquad A(4, 1) = 2^{16} - 3$$

$$A(4,2) = A(3,65533) = 2^{65536} - 3 \approx 2 \times 10^{19728}$$
$$A(4,3) \approx A(3, 210^{19728}) \approx 10^{6 \times 10^{19727}}$$

アッカーマン関数は 2 重再帰関数任意の原始再帰関数 $f(x) < A(c,x)$ となる整数 $c$ が存在する。

# 関数の強さ

原始再帰関数 $<$ 2 重再帰 $< \cdots <$n 重再帰

- 計算可能関数（グラハム数、アッカーマン関数）
- 計算不可能関数（ビジービーバー、$F_4$）
- 理論体系の強さと結びついた巨大数

# 強さを比較する物差し

順序数階層：巨大数を比較する手段

ハーディー階層 細やかな近似に向く

急増増加関数 FGH 大まかな近似に使われる

# 順序数

自然数の拡張としてカントールが提唱
順番に並べられたものに番号を振るための"数"
無限個あるものに対してもその"順序"を数えることができる

**Definition 定義 (順序数)**

整列集合の順序型

順序集合 $(A, \leq)$,$A = \{1, 2, 3\}$ 順序数は 3
順序集合 $(\mathbb{N}, \leq)$,$\mathbb{N} = \{1, 2, 3, \ldots\}$ 極限順序数 $\omega$

**Definition 定義 (ノイマンによる順序数の定義)**

自分自身よりも小さい全ての順序数の集合

$$
\begin{aligned}
0 &= \{\} \\
1 &= \{0\} \\
2 &= \{0, 1\} \\
&\vdots \\
\omega &= \{0, 1, 2, \dots\} \\
\omega + 1 &= \{0, 1, 2, \dots, \omega\} \\
\omega + 2 &= \{0, 1, 2, \dots, \omega, \omega + 1\} \\
&\vdots \\
\omega \times 2 = \omega + \omega &= \{0, 1, 2, \dots, \omega, \omega + 1, \dots\}
\end{aligned}
$$

# 急増増加関数 FGH(Fast-Growing Hierarchy)

$$\omega \times 3 = \{\omega \times 2 + 1, \omega \times 2 + 2, \ldots\}$$
$$\omega \times 4 = \{\omega \times 3 + 1, \omega \times 3 + 2, \ldots\}$$
$$\vdots$$
$$\omega^2 = \omega \times \omega = \{0, \omega, \omega \times 2, \ldots\}$$

**Definition 定義 (急増増加関数 (Fast-Growing Hierarchy))**

$$F[0](n) = n + 1$$
$$F[\alpha + 1](n) = F[\alpha]^n(n)$$
$$F[\alpha](n) = F[\alpha_n](n)(\alphaが極限順序数のとき)$$

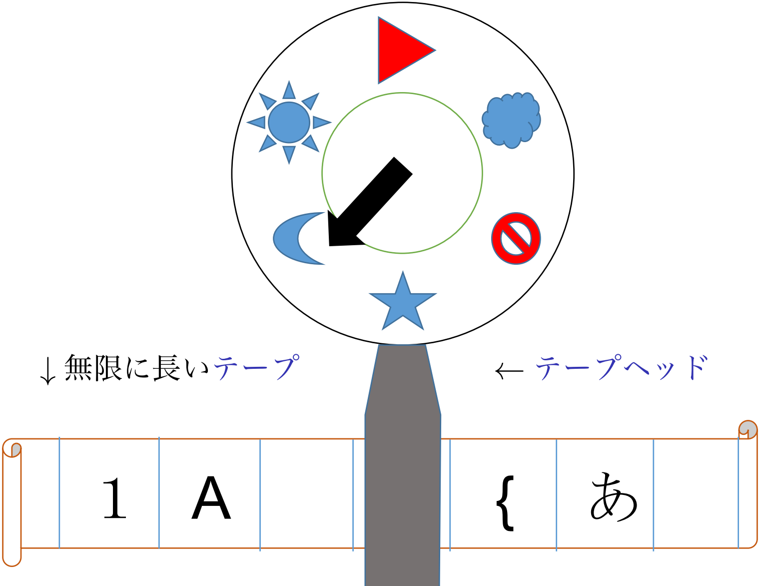
順序数の階層構造 $\rightleftharpoons$ 再帰性の回数

$f_n = 3 \uparrow^n 3$ としたとき

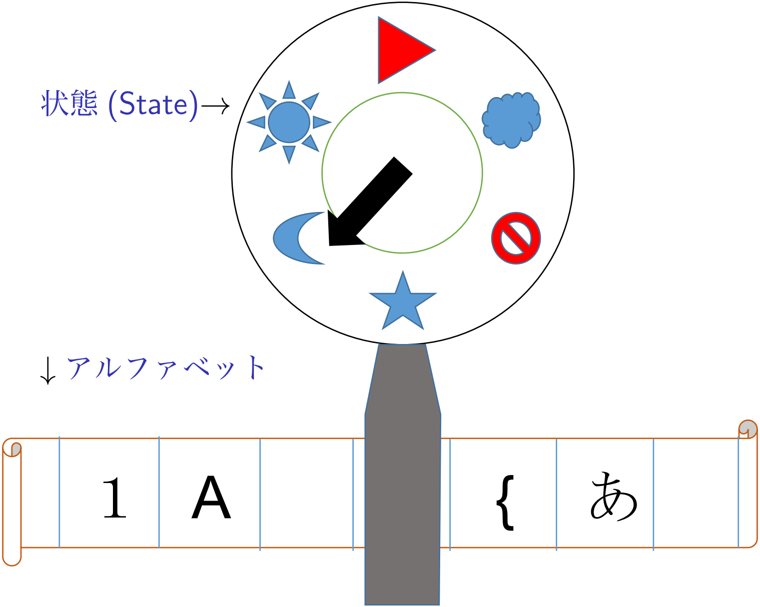
$$\text{グラハム数} \approx F[\omega + 1](64) = f^{64}(64)$$

$$\text{アッカーマン関数} A(n, n) < F[\omega](n)$$

# チューリングマシン



## チューリングマシン



# チューリングマシンの構成

テープ 無限に長いテープ。1 セルに 1 文字書き込み

ヘッド セルの読み込み、書き換え、左右 1 セル移動

状態 (state) 機械の中で記録できる記号の有限集合：表記 $S$

- 初期状態 (initial state) 動き始めの状態
- 停止状態 (halting state) この状態になると停止
- その他の状態

アルファベット テープに読み書きできる記号の有限集合

遷移関数 (Transition Function) ヘッドと状態の動きを定める

# 遷移関数：一進数で引き算する遷移関数

| 遷移前状態 | 読み込み | 遷移後状態 | 書き込み | 移動方向 |
|---|---|---|---|---|
| ♣ | space | ♣ | space | → |
| ♣ | 1 | ♣ | 1 | → |
| ♣ | — | ♣ | - | → |
| ♣ | = | ♣ | space | ← |
| ♢ | 1 | ♡ | = | ← |
| ♢ | — | Halt | space | n/a |
| ♡ | 1 | ♡ | 1 | ← |
| ♡ | — | ♠ | - | ← |
| ♠ | space | ♠ | space | ← |
| ♠ | 1 | ♣ | space | → |

# チューリングマシンの性質

- 単純さと理解のしやすさがちょうど良い
- 万能チューリングマシン (Universal Turing machine)

→ ほぼ全ての計算システムを模倣可能
チューリング完全チューリングマシンと同じ能力を計算機が持つこと。機械仕掛けの計算機が持つ能力の最大の１つ

# チューリングマシンの性質

- 単純さと理解のしやすさがちょうど良い
- 万能チューリングマシン (Universal Turing machine)

→ ほぼ全ての計算システムを模倣可能
チューリング完全チューリングマシンと同じ能力を計算機が持つこと。機械仕掛けの計算機が持つ能力の最大の１つ

## マリオメーカーもチューリング完全！

停止問題 (Halting Problem) 本質はゲーデルの不完全性定理と同じ

**Theorem 定理 (ゲーデルの不完全性定理)**

ペアノ算術を表現できる程度に強力な論理体系では、証明もその否定の照明もできない言明がある。

**停止問題**

プログラム $P$ が与えられた時、$P$ がいつ停止するかを与えるプログラムを書くことはできるか？

**Definition 定義 (実行的計算装置 (effective computing device:ESC))**

$$C : N(\text{プログラム}) \times N(\text{プログラムの入力}) \to N(\text{プログラムの結果})$$

プログラムが停止しないなら ESC は何も返さない（定義域に含まれていないと言う）
プログラム $f$ の入力 7 に対して

- 停止するとき $C(f, 7) = 4$
- 停止しない $C(f, 7) = \perp$

適当な対関数 $\mathrm{pair}(x, y)$ を用いて

**Definition 定義 (停止性神託)**

$$\forall P, \forall i : C(O, \mathrm{pair}(p, i)) = \begin{cases} 0 & C(P, i) = \perp \text{のとき} \\ 1 & C(P, i) \neq \perp \text{のとき} \end{cases}$$

「このような神託プログラム $O$ が存在するか？」が停止問題

適当な対関数 $\mathrm{pair}(x, y)$ を用いて

**Definition 定義 (停止性神託)**

$$\forall P, \forall i : C(O, \mathrm{pair}(p, i)) = \begin{cases} 0 & C(P, i) = \perp \text{のとき} \\ 1 & C(P, i) \neq \perp \text{のとき} \end{cases}$$

「このような神託プログラム $O$ が存在するか？」が停止問題
Ans. そのような神託は存在しない
美味しい証明があるのですが省きます…

# ビジービーバー関数

1962 年に数学者 Tibor Radó が提唱

**Definition 定義 (ビジービーバーゲーム (busy beaver game))**

$n$ 状態 2 記号 (0,1) のチューリングマシン
テープは全て 0 からスタート
もし、マシンが停止 → テープ上の「1」の数を返す
$\forall n$ 状態 2 記号のマシンの中で最大の「1」の数を返すもの
$=n$ 状態のビジービーバー
優勝マシン … 今の所のマシンの中で最大の値を返すもの

優勝マシンがビジービーバーなのか？
全てのマシンが停止しないと分からない ← 停止問題

**Definition 定義 (ビジービーバー関数)**

ビジービーバー関数 $\Sigma(n)$ : $\forall n$ 状態 2 記号のマシンがビジービーバーゲームで返す値の中で最大のもの

定義はできるが計算不可能な関数
※「C 言語において 10 文字で書ける最大の数」のような問題もビジービーバーに帰着される

# 神託機械

**Definition 定義 (神託機械 (oracle machine))**

計算不可能な工程を飛ばしてその答えだけをだす関数、神託の存在を仮定する。神託を含むチューリングマシンを神託機械という。

# ふぃっしゅ数 Ver.4($F_4$)

**Definition 定義 (ふぃっしゅ数 Ver.4($F_4$))**

1. 関数 $f$ から関数 $g$ への写像 $s'(1)$ を以下で定める
関数 $f$ を神託として持つチューリングマシンの神託機械を考え、このマシンによるビジービーバー関数を $g$ とする。
2. $s'(n)(n>1)$ 及び $ss'(n)(n>0)$ を、$F_3$ と同様に定める。すなわち、

$$s'(n)f = g \quad ; \quad g(x) = [s'(n-1)^x]f(x)(n>1)$$

$$ss'(1)f = g \quad ; \quad g(x) = s'(x)f(x)$$

$$ss'(n)f = g \quad ; \quad g(x) = [ss'(n-1)^x]f(x)(n>1)$$

## Definition 定義 (ふぃっしゅ数 Ver.4($F_4$))

3. ふぃっしゅ関数 $F_4$ を以下のように定める。

$$F_4(x) = ss'(2)^{63}f \quad ; \quad f(x) = x + 1$$

4. ふぃっしゅ数 $F_4 = F_4^{63}(3)$ とする。

$$\begin{bmatrix} m \\ f(x) \end{bmatrix} \xrightarrow{s\text{ 変換}} \begin{bmatrix} n \\ g(x) \end{bmatrix}$$

$$\begin{bmatrix} m \\ f(x) \\ s\text{ 変換} \end{bmatrix} \xrightarrow{ss\text{ 変換}} \begin{bmatrix} n \\ g(x) \\ s2\text{ 変換} \end{bmatrix}$$

$F_1$ はアッカーマン関数、$F_2$ は ss 変換の再帰性を上げる . . .

# ふぃっしゅ数 Ver.4($F_4$)

**Definition 定義 (ふぃっしゅ数 Ver.4($F_4$))**

1. 関数 $f$ から関数 $g$ への写像 $s'(1)$ を以下で定める
関数 $f$ を神託として持つチューリングマシンの神託機械を考え、このマシンによるビジービーバー関数を $g$ とする。
2. $s'(n)(n > 1)$ 及び $ss'(n)(n > 0)$ を、$F_3$ と同様に定める。すなわち、

$$s'(n)f = g \quad ; \quad g(x) = [s'(n-1)^x]f(x)(n > 1)$$

$$ss'(1)f = g \quad ; \quad g(x) = s'(x)f(x)$$

$$ss'(n)f = g \quad ; \quad g(x) = [ss'(n-1)^x]f(x)(n > 1)$$

# $F_4$ の近似

$$F_4 \approx F[\omega^{\omega+1} \times 63](n)$$

- Mark C.Chu-Carroll 著 cocoatomo 訳 『グッド・マスギークのための数・論理・計算機科学』（オーム社,2016）
- ふぃっしゅっしゅ（フィッシュ）『巨大数論　2 版 $\beta$』(2016,2017)
- Michael A. Nielsen, Isaac L. Chuang 共著、木村達也訳『量子コンピュータと量子通信 I 量子力学とコンピュータ科学』（オーム社,2004）
- 小林銅蟲著　『寿司　虚空編　-Sushi Kokuu Hen-』（pixiv,2013）
- 松坂和夫著　『集合・位相入門』(岩波書店, 1968)