文字列写像による順序数への対応

５年Ａ組　竹内　伶河
指導教員　川口　慎二

１．要約

文字列によって自然数関数を定義することで、その文字列関数と順序数階層を対応させる。そうすることで文字列と順序数との対応を考えることができる。そして、証明論等と文字列の対応を試みる。

キーワード　文字列写像、順序数、ハーディ階層、バッハマン・ハワード順序数

２．研究の背景と目的

巨大数を得るためには、強い関数を用意するのが適切であるとわかり、文字列と関数を対応させることにより強い関数を作ることを目指した。更に文字列と順序数との対応を考え、それが証明論と対応するのではないかと思い、順序数に近い形で文字列をより強くすることを試みた。

３．研究内容

３－１　ハーディ階層

関数の強さを評価するためにハーディ階層を導入する。そのためにまず、順序数の基本列を定義する。

<u>定義１</u>

共終数が ω である極限順序数 α に対して、α に収束する順序数の単調増加列を、α の**基本列**という。

例えば､順序数 ω の基本列は $0, 1, 2, 3, \cdots$ であり、順序数 $\omega+\omega$ の基本列は

$$\omega+1,\ \omega+2,\ \omega+3,\ \omega+4,\ \omega+5,\ \cdots$$

である。また、順序数 ω^{ω} の基本列は

$$\omega^1,\ \omega^2,\ \omega^3,\ \omega^4,\ \omega^5,\ \cdots$$

である。

ハーディ階層とは、順序数 α に対して、関数 $H_\alpha : \mathbb{N} \to \mathbb{N}$ を定める順序数による関数の階層 $\{H_\alpha(n)\}$ のことをいい、関数の大きさを評価したり、比較したりするときに用いられる。数学的には以下のように定義される。

<u>定義２</u>

α を任意の順序数、n を任意の自然数とする。また、$\beta[n]$ を極限順序数 β の基本列の n 番目とする。このとき、関数の階層 $\{H_\alpha(n)\}$ を次のように定義する。

$\alpha = 0$ のとき、$H_\alpha(n) = n$

α が後続順序数のとき、　$H_\alpha(n) = H_{\alpha-1}(n+1)$

α が極限順序数のとき、

$$H_\alpha(n) = H_{\alpha[n]}(n)$$

ここで、最小の超限順序数である ω から有限回の加算や乗算、冪乗では到達できない最小の超限順序数を ε_0 と表し、**イプシロン・ノート**または**イプシロン・ゼロ**とよぶ。$\varepsilon_0 = \omega^{\omega^{\omega^{\cdots}}}$ とも表現される。

ε_0以下の極限順序数αの基本列を定める方法としてワイナー階層とよばれるものがある。

<u>定義 3</u>

αを$\alpha \le \varepsilon_0$である極限順序数とする。このとき、αの基本列$\{a[n]\}$を以下のように帰納的に定義する。

$\alpha = \omega$のとき、$\omega[n] = n$

$\alpha = \omega^{\alpha}$のとき、$\omega^{\alpha}[n] = \omega^{\alpha[n]}$

$\alpha = \omega^{\alpha+1}$のとき、$\omega^{\beta+1}[n] = \omega^{\beta} n$

$\alpha = \omega^{\beta}$ かつβが極限順序数のとき、

$$\omega^{\beta}[n] = \omega^{\beta[n]}$$

$\alpha = \omega^{\gamma_1} + \omega^{\gamma_2} + \cdots + \omega^{\gamma_{k-1}} + \omega^{\gamma_k}$，ただし、$\gamma_1 \ge \gamma_2 \ge \cdots \ge \gamma_{k-1} \ge \gamma_k$ （カントール標準形）のとき、

$$(\omega^{\gamma_1} + \omega^{\gamma_2} + \cdots + \omega^{\gamma_{k-1}} + \omega^{\gamma_k})[n]$$
$$= \omega^{\gamma_1} + \omega^{\gamma_2} + \cdots + \omega^{\gamma_{k-1}} + (\omega^{\gamma_k}[n])$$

$\alpha = \varepsilon_0$のとき、

$$\varepsilon_0[0] = 1 \text{ かつ } \varepsilon_0[n+1] = \omega^{\varepsilon_0[n]}$$

この基本列$\{a[n]\}$をワイナー階層という。

ε_0よりも大きい順序数を定義する方法の一つとして、ヴェブレン関数がある。

<u>定義 4</u>

$\alpha, \beta, \gamma, \delta$を任意の順序数とする。このときヴェブレン関数$\{\phi_{\alpha}(\beta)\}$を以下のように定義する。

・$\alpha = 0$のとき、$\phi_{\alpha}(\gamma) = \omega^{\gamma}$

・$\alpha \ne 0$のとき、$\phi_{\alpha}(\gamma)$はαより小さいすべての順序数βに対して$\phi_{\beta}(\delta) = \delta$が成り立つ順序数のうち$\beta$番目

この関数によってε_0より大きな順序数を表すことができ、ハーディ階層でより強い関数を表すことができるようになる。また、ヴェブレン関数における$\phi_1(0)$は、ε_0と等しくなる。

３－２　表記の定義

[1]において、写像を用いて関数を強化する方法を考案した。[2]では、文字列によって正整数関数を定め、順序数と１対１対応させた。まず、表記の更なる拡張のため、今回は[2]の定義を改良した。

$$\text{空列} \in S$$
$$X \in S \wedge Y \in S \Rightarrow XY \in S$$
$$X \in S \Rightarrow (X) \in S$$

を満たす最小の文字列集合Sと自然数集合$\mathbb{N}$の直積$S \times \mathbb{N}$を始域とし、Sを終域とする写像fを以下のように定義する。ただし、

$X, Y, Z \in S$, $Z \ne$空列とする。

$$f[X(), n] = X$$
$$f[X(Y()), n] = X\underbrace{(Y)(Y)\cdots(Y)}_{n}$$
$$f[X(Y(Z)), n] = X(f[Y(Z), n])$$

いま、A は$*|$を含む文字列の集合、B は$()$のみの集合、C, D を$()\{\}*|$による任意の文字列の集合と定義する。ここで、$*|$はAにおける演算であり、$|$は最も優先度の低い演算である。さらに、

$$|A|=A$$
$$A*()=A$$
$$A*|A()|=|A*A|A$$
$$(A())=(A)*A$$

であり、

$$A,\ B,\ C\in S$$
$$A\in S\Rightarrow\{A\}\in S$$
$$A\in S\Rightarrow A*A\in S$$
$$A\in S\Rightarrow|A|\in S$$
$$\text{空列}\in S$$

である。このとき、自然数mに対して、$f(m)$を次のように定義する。まず、$f(0)$はfと同義である。kを正整数とする。

$$f(0)[B,\ n]=B$$
$$f(k+1)[B,\ n]=(f(k)[B,\ n])$$
$$f(0)[(\{\}),n]=f(0)[f(n)[(),\ n],\ n]$$

$C\{\}D$ における$\{\}$は最も右のものである（$C\{\}D\in S$ である）。

$$f(0)[C\{\}D,\ n]=f(n)[C\{\}D,\ n]$$
$$f(k+1)[C\{\}D,\ n]$$
$$=f(0)[Cf(k)[C\{\}D,\ n]D,\ n]$$
$$f(1)[C\{\}D,n]=f(0)[C()D,\ n]$$

以上の文字列と自然数から自然数への写像gを以下のように定義する。

$$g[\text{空列},\ n]=n$$
$$g[X,\ n]=g[f[X,\ n],\ n+1]$$

このように定義した表記の$g[B,\ n]$の最大はハーディ階層として$H_{\varepsilon_0}(n)$となる。

また、$g[(\{(())\}),\ n]$は$H_{\phi_\omega(0)}(n)$と対応する([2])。そして、Aは$BHO\varphi_0(\varepsilon_{\Omega+1})$に対応すると予想している。ここで、$BHO\varphi_0(\varepsilon_{\Omega+1})$とはバッハマン・ハワード順序数(Bachmann-Howard ordinal)と呼ばれる順序数である([3])。

３－３　別のアプローチ

３－２節で用いた表記は、括弧を多重にネストする構造を取ることで順序数関数のネストとの対応を試みたものだが、別のアプローチとして原始数列を参考にしたものを以下のように考案した。

$$\text{空列}\in S$$
$$X\in S\wedge Y\in S\Rightarrow XY\in S$$
$$X\in S\Rightarrow(X)\in S$$

を満たす最小の文字列集合Sと自然数集合$\mathbb{N}$の直積$S\times\mathbb{N}$を始域とし、Sを終域とする写像Fを以下のように定義する。ただし、$X,\ Y,\ Z\in S,\ Z\neq\text{空列}$とする。

まず、gはgood partを，bはbad partを表すものとする。

$i<k$ かつS_kより弱く、最も強いS_i（強い文字列とは、より大きい順序数と対応する文字列を指す）となる最大の非負整数iが存在するとき、

$$g=(S_0,\ ...,\ S_{i-1}),\ \ b=(S_i,\ ...,\ S_{k-1})$$

であるとし、i が存在しないとき b は空列である。さらに、

$$F[X(), n] = X$$

$$F[[gbS_i], n] = F[[g\underbrace{bb\cdots b}_{n}], n]$$

$$F[X(Y(Z)), n] = X(F[Y(Z), n])$$

とする。

以上の文字列と自然数から自然数への写像 G を以下のように定義する。

$$G(\text{空列}, n) = n$$

$$G(X, n) = G(F(X, n), n+1)$$

この表記はハーディ階層で $\phi_2(0) = \zeta_0$ と対応すると考えられる。そして、このアプローチは[1]の定義4から着想を得た。つまり、文字列 S_n を並べたまとまり $[S_1,\ldots,S_k]$ を複数並べることが可能である。

また、$[S_1,\ldots,S_k]$ において文字列を自然数個ではなく、個数と文字列を以下のように対応させることができる。

S,T を任意の文字列の列とするとき、

$$f[[S][T,X],n] = f[f[S,n][T,X,Y],n]$$

$$[][S] = [S]$$

[1]のように文字列の列をネストさせたり、反復合成冪を取ったりしたわけではないが、列の個数を順序数に対応させたためより大きい順序数に対応させられると期待できる。

３－４　更なる拡張

３－２節における写像 $f(m)$ において、m には自然数を入力したが、文字列を入れることで更なる拡張を試みる。そこで、$f(A)$ を定義する。$(\underbrace{\{\{\cdots\{}_{n}\underbrace{\}\}\cdots\}}_{n})$ を α とおく。

$$f()[B, n] = B$$

$$f(A())[B, n] = (f(A)[B, n])$$

$$f()[(\{\}), n] = f()[f(\alpha)[(), n], n]$$

$C\{\}D$ における $\{\}$ は最も右のものである。

$$f()[C\{\}D, n] = f(\alpha)[C\{\}D, n]$$

$$f(A())[C\{\}D, n] = f()[Cf(A)[C\{\}D, n]D, n]$$

$$f(())[C\{\}D, n] = f()[C()D, n]$$

このようにすることで、$f(A)$ の A に $f[A, n]$ を入れることができるようになり、ネスト構造を取ることができる。

さらに、３－３節の定義を組み合わせれば、$f(A)$ のネストの重ねる回数と文字列を対応させることができるのではないかと考える。そして、これを新たな括弧の導入に利用できると期待できる。こうすることにより拡張ブーフホルツ順序数 *EBO* (Extended Buchholz Ordinal)との対応を目指すことができるのではないかと考えている。

４．今後の展望

表記も複雑になり、この大きさの順序数となると定義が複雑になるため、対応について整理したい。

また、３－４節の通り、さらにネストする箇所を作ることで新たな括弧を定義できると期待できるため試みたい。加えて、今回は証明論との対応についてわかったことが少ないため、その解析も行いたい。

５．参考文献

[1] 竹内　伶河，「関数を強くする」，2020

年度奈良女子大学附属中等教育学校
SSH 生徒研究論文集,
http://www2.nara-wu.ac.jp/fuchuko/media/sites/11/20_5.pdf
[2] 竹内　伶河,「文字列によって得られる強い関数」, 2021 年度奈良女子大学附属中等教育学校 SSH 生徒研究論文集,
https://nwuss.nara-wu.ac.jp/media/sites/11/ssh21_2.pdf
[3] 極限順序数一覧
https://googology.fandom.com/ja/wiki/%E6%A5%B5%E9%99%90%E9%A0%86%E5%BA%8F%E6%95%B0%E3%81%AE%E4%B8%80%E8%A6%A7