

巨大数の樹海へようこそ！ *

木原 貴行

名古屋大学 情報学部・情報学研究科

最終更新日: 2021 年 6 月 1 日

目次

1	TREE(3) と呼ばれる超巨大数	2
2	TREE(3) はどれくらい大きいのか？	13
3	TREE(3) はなぜ有限なのか？	45
4	あとがき	65

* 本講義ノートは、2021 年度春 1 期開講の名古屋大学情報学部における講義「数理情報学序論 1」のための講義資料である。

§ 1. TREE(3) と呼ばれる超巨大数

1.1. まえがき

TREE(3) と呼ばれる巨大な数の話をしよう。これは、組合せ論におけるクラスカルの木の定理 (*Kruskal's tree theorem*) と呼ばれる定理の研究の過程で発見された、途方もなく巨大な数である。数学における巨大な数というと、グラハム数が有名であり、これもまたラムゼー理論と呼ばれる組合せ論の一分野で用いられたものである。グラハム数は

「数学の証明で使われたことのある中で最も大きい数」

としてギネス記録になっていることで広く知られている。しかし、そのギネス記録のグラハム数など比べ物にならないほど巨大な数が、TREE(3) である。

グラハム数 (ギネス記録) \ll TREE(3).

特にギネス記録申請していないためなのかどうか理由は知らないが、TREE(3) は残念ながらギネスブックには載っていない。とはいえ、ある問題の解の上界として与えられたに過ぎないグラハム数よりは、ある数学的命題の厳密値として得られる TREE(3) の方が、数学的に本質的な数であるとも言える。

余談であるが、TREE(3) はギネスブックには載っていないものの、TREE(3) の発見者であるハーヴェイ・フリードマン (Harvey Friedman) は別のギネス記録を所持している。青年フリードマンは、18 歳の時点で既にマサチューセッツ工科大学において数学の博士号を取得し、直後にスタンフォード大学の哲学科の助教授となった。これにより、フリードマンは「世界最年少・18 歳教授」という当時のギネス記録となったのである。

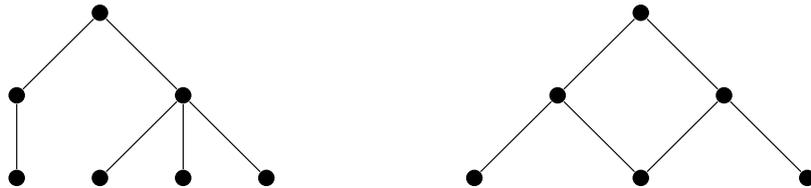
とにかく、数学の話に戻ろう。グラハム数はあからさまに巨大数として定義されるが、TREE(3) を一見して巨大な数だと気づく人はほとんどいないだろう。TREE(3) は極めて簡単に定義されたごく普通の数を装っているが、実は、この世のほとんどの人間が想像だにしたことのない途方もなく巨大な数となる。この事実は、数多くの人々に衝撃を与えた。この巨大数 TREE(3) が有名になった切っ掛けは、おそらく Numberphile という YouTube チャンネルが 2017 年 10 月に投稿した TREE(3) に関する動画^{*1}であり、2020 年 4 月現在において、この動画は 91 万回以上再生されている。無論、これから語る数に比べれば、91 万など微小な数であるが。

さて、巨大数 TREE(3) は、ただただとてつもなく巨大な数、というだけではなく、その背後で様々な数学の分野 (組合せ論, 順序数論, 証明論, 抽象代数など) が入り混じっており、学術的にも非常に興味深い数である。本講義では、この巨大数 TREE(3) の面白さを伝えることを目指していきたい。

^{*1} Numberphile, The Enormous TREE(3), <https://www.youtube.com/watch?v=3P6DWAwwViU>

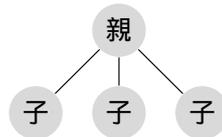
1.2. グラフ理論の基本用語

さて、それでは本題の TREE(3) とは一体なんであろうか。TREE というからには、「木」に関する数である。これを説明するために、まずはグラフ理論の用語から導入する必要がある。グラフ理論において、木 (*tree*) とは、閉路を持たない連結な無向グラフを意味する。



たとえば、上図の左は木であるが、右はループを持つので木ではない。本稿では、木と言った場合に、空でない有限の根付き木 (*rooted tree*) を表すことにする。つまり、頂点数は 1 以上だが有限であり、常に根 (root) と呼ばれる頂点が 1 つ指定されている。第 1 節では、木を描くときに、木の根は必ず最上位に配置することにする。

木の各頂点はしばしばノード (node) と呼ばれる。辺で繋がったノードについて、根に近い側を親ノードと呼び、根から遠い側を子ノードと呼ぶ



ノード σ と τ に最も近い共通の祖先は $\sigma \wedge \tau$ と書かれる。この $\sigma \wedge \tau$ は、 σ と τ の下限 (*infimum*) と呼ばれる。



ノード σ と τ について、上図の青線で描かれるように、 σ, τ からどんどん親ノードに遡ってゆき、最初の共通祖先となるノードが下限 $\sigma \wedge \tau$ である。上の右図においては、ノード γ もまたノード σ と τ の共通祖先であるが、より近い共通祖先があるので、 γ は σ と τ の下限ではない。

また、 σ が τ の祖先であるとき、 $\sigma \sqsubseteq \tau$ と書く。 $\sigma \sqsubseteq \tau$ または $\sigma = \tau$ であるとき、 $\sigma \sqsubseteq \tau$ と書く。これは、 $\sigma \wedge \tau = \sigma$ となっていることと同値である。

$$\sigma \sqsubseteq \tau \iff (\sigma \sqsubset \tau \text{ or } \sigma = \tau) \iff \sigma \wedge \tau = \sigma.$$

根付き木のグラフ構造は、木のノードの集合 T と上記のデータ \sqsubseteq から一意に復元できる。このため、以後は、根付き木とは対 (T, \sqsubseteq) を指すことと考える (これは、根付き木を有向グラフと考え

ることと同等である)。ただし、しばしば \sqsubseteq は省略して、単に T と書く。3 つ組 (T, \sqsubseteq, \wedge) は下半束 (*lower semilattice*) と呼ばれるものになっているが、詳細は省略する。

下半束埋め込み: 本稿において鍵となる概念が、下半束埋め込み、略して \wedge -埋め込みと呼ばれるものである。関数 $h: S \rightarrow T$ が単射 (*injective*) であるとは、

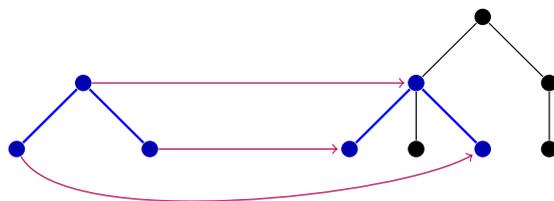
$$(\forall \sigma, \tau \in S) [\sigma \neq \tau \implies h(\sigma) \neq h(\tau)]$$

であることを意味する。木 S, T 上の関数 $h: S \rightarrow T$ が下半束の準同型、略して \wedge -準同型 (*inf-homomorphism*) とは、順序 \sqsubseteq と下限 \wedge を保つことを意味する。つまり、以下の 2 条件を満たすことを意味する。

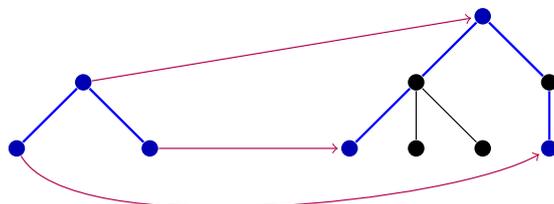
$$\begin{aligned} (\forall \sigma, \tau \in S) [\sigma \sqsubseteq \tau \implies h(\sigma) \sqsubseteq h(\tau)], \\ (\forall \sigma, \tau \in S) h(\sigma \wedge \tau) = h(\sigma) \wedge h(\tau). \end{aligned}$$

単射であるような \wedge -準同型のことを、下半束埋め込み、略して \wedge -埋め込み (*inf-embedding*) と呼ぶ。 \wedge -埋め込みの例をいくつか見ていこう。以下、紫矢印によって、左の木 S から右の木 T への関数 $h: S \rightarrow T$ を表す。

例 1.1. 次は \wedge -埋め込みの例であるが、下限を保つことが重要なので、左右の位置は入れ替わっても問題はないという例になっている。

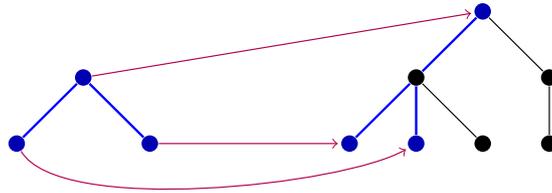


例 1.2. 次もまた \wedge -埋め込みの例であるが、必ずしも隣接ノードを隣接ノードに移す必要はないという例になっている。

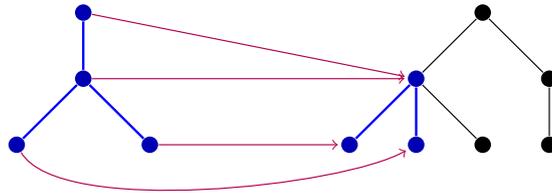


例 1.3. 次は、 \wedge -埋め込みではない例である。順序 \sqsubseteq を保つ単射となっているが、下限 \wedge を保つ

ていない。



例 1.4. 少し左の木を変化を加えてみよう．次は， \wedge -準同型であるが， \wedge -埋め込みではない例である．つまり，木を潰してしまっており，単射になっていない．



定義 1.5. 木 S, T について， S が T に \wedge -埋め込み可能 (*inf-embeddable*) であるとは， \wedge -埋め込み $h: S \rightarrow T$ が存在することを意味する． S が T に \wedge -埋め込み可能であるとき， $S \leq T$ と書く．

たとえば，例 1.4 の紫矢印で表される関数は， \wedge -埋め込みではないが，左の木は右の木に \wedge -埋め込み可能である．具体的には，左の木の根の行き先を右の木の根に変えればよいだけである．

1.3. 埋め込み不可能列の長さ

この \wedge -埋め込み可能性に関する重要な定理が，クラスカルの木の本の定理と呼ばれるものである．歴史的背景やその証明は第 3 節以降で述べるが，クラスカルの本の定理とは，次の主張である．

定理 1.6 (クラスカルの本の定理). 次の条件を満たす木の無限列 T_1, T_2, T_3, \dots は存在しない．

$$(\forall m, n) [m < n \implies T_m \not\leq T_n]. \quad (1)$$

式 (1) は，つまり，木 T_1, T_2, T_3, \dots を並べていったとき，どの木も手前の木を \wedge -埋め込み不可能であることを意味する．そして，クラスカルの本の定理は，無限長の \wedge -埋め込み不可能列は存在しないことを述べる．つまり，どのように木 T_1, T_2, T_3, \dots を並べていっても，どこかのタイミングで必ず，手前の木を \wedge -埋め込み可能であるような木 T_n が現れる．このように， \wedge -埋め込み不可能列の長さが有限であることは保証されるのであるが，しかし，有限とはいっても実際にどれくらいの長さになり得るか，ということ調べるのは重要である．

これを見積もるために、木 T_i の頂点数を高々 $n + i$ に制限することにしよう。この制限の下で、最長の \wedge -埋め込み不可能列の長さを $\text{tree}(n)$ と書くことにする。

定義 1.7. 次の条件を満たす木の列 T_1, T_2, \dots, T_ℓ が存在するような最大の ℓ を $\text{tree}(n)$ と表す。

1. 任意の $1 \leq j < k \leq \ell$ について、 $T_j \not\preceq T_k$ である。
2. 任意の $1 \leq i \leq \ell$ について、 T_i の頂点数は高々 $n + i$ である。

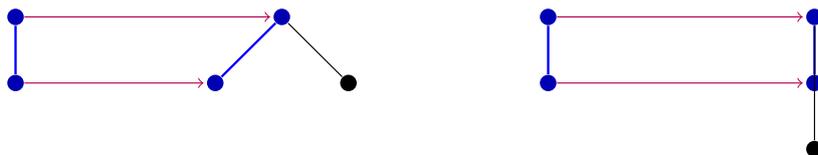
具体的に、 $n = 0, 1, 2$ のとき $\text{tree}(n)$ を計算してみよう。 $n = 0$ のときには、 T_i は頂点数 $\leq i$ の木を選ばなければならない。しかし、頂点数 1 の木と頂点数 2 の木はそれぞれ一種類しか存在せず、たとえば以下のように \wedge -埋め込み可能である。



実際、頂点数 1 の木は、いかなる木にも \wedge -埋め込み可能である。したがって、長さ 2 の \wedge -埋め込み不可能列は存在しないから、

$$\text{tree}(0) = 1$$

である。つづいて、 $n = 1$ のときであるが、 T_i は頂点数 $\leq i + 1$ の木を選ばなければならない。しかし、初手の T_1 として頂点数 1 の木を選んでしまうと、先ほどと同様にして次の手を打てないから、初手 T_1 は頂点数 2 の木を選ぶ必要がある。頂点数 3 の木は 2 種類あるが、どちらも頂点数 2 の木を \wedge -埋め込み可能である。



したがって、第 2 手で頂点数 3 の木を打つことはできない。頂点数 2 の木についても同様であるから、第 2 手として可能なのは頂点数 1 の木のみである。

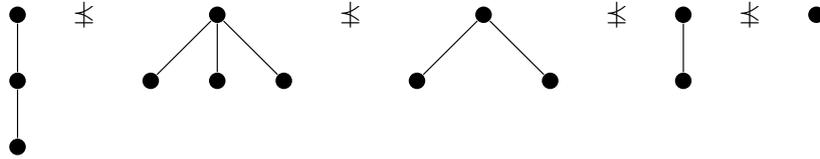


先ほど見たように、頂点数 1 の木は任意の木に \wedge -埋め込み可能であるから、これが限界である。したがって、

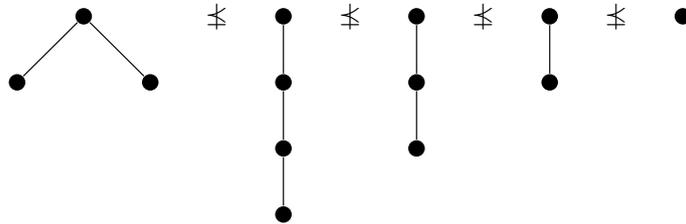
$$\text{tree}(1) = 2$$

を得る。つづいて、 $n = 2$ のときであるが、 T_i は頂点数 $\leq i + 2$ の木を選ばなければならない。したがって、初手 T_1 として、頂点数 3 の木を選ぶことができる。初期頂点数 3 の \wedge -埋め込み不可能

列の具体例を挙げよう.



頂点数は 3, 4, 3, 2, 1 なのでルール (定義 1.7 の条件) を遵守している. 上図について, 左から右への \wedge -埋め込み不可能性 \perp は, 隣接していないものについても確認しなければならないことに注意しよう. つまり, 上図中の記号 \perp は両隣の木だけに掛かっているのではなく, 両側すべての木に対する主張である. 次もまた初期頂点数 3 の \wedge -埋め込み不可能列の具体例である.

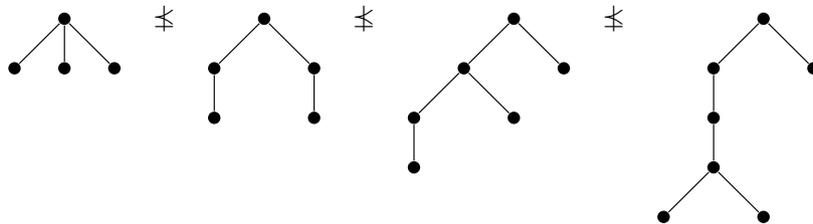


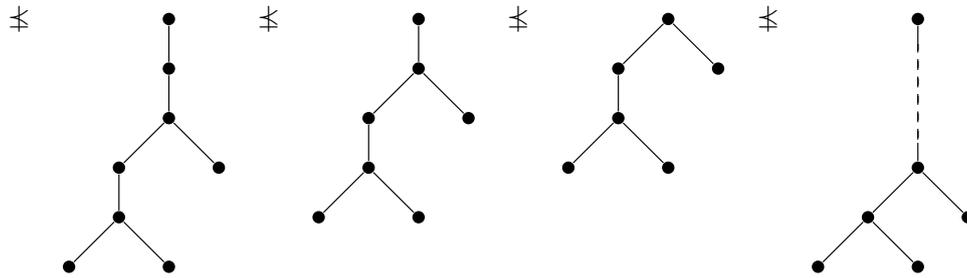
これについても, 頂点数は 3, 4, 3, 2, 1 なのでルール (定義 1.7 の条件) は遵守している. 両方も長さ 5 の列であるが, 実はこれが最長である. つまり, 次を得る.

$$\text{tree}(2) = 5.$$

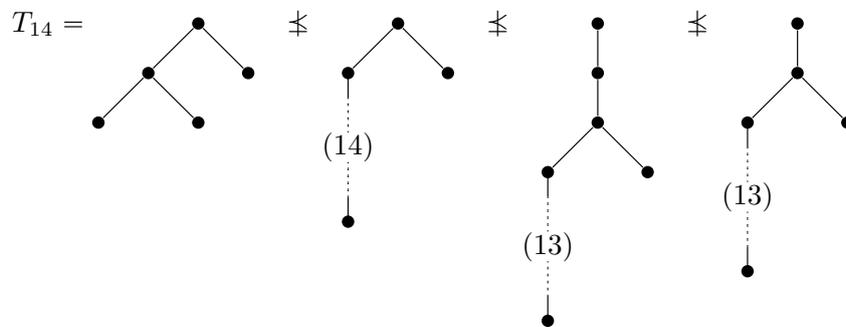
問題 1. $\text{tree}(2) = 5$ であることの厳密な証明を与えよ. つまり, 定義 1.7 のルールを遵守する初期頂点数 3 の \wedge -埋め込み不可能列で, 長さ 6 以上のものは存在しないことを証明せよ.

それでは, つづいて $\text{tree}(3)$ も計算してみよう.....と言いたいところであるが, この辺りから tree の値は爆発的に膨れ上がるので, そう簡単に求めることはできない. しかし, それが大きいものになりそうだという感覚を掴むために, たとえば, 初期頂点数 4 の \wedge -埋め込み不可能列として, 以下のようなものを考えてみよう.





ここで、上図における右下の木 T_8 について、根から最初の分岐の間には 5 つの頂点が省略されている。上図の木の頂点数は順に 4, 5, 6, 7, 8, 7, 6, 11 であるから、ルールを遵守している。 T_8 から 6 ステップ掛けて、根から順に頂点を 1 つずつ落としていこう。このとき、 T_{14} において、下図左のようになり、さらに \wedge -埋め込み不可能列の作成を進めていく。



ここで (14) と書いている部分について、そこに 14 個の頂点が省略されていることを意味する。(13) と書いた部分についても同様である。よって、頂点数は順に 5, 18, 19, 18 であるから、ルールを遵守している。この \wedge -埋め込み不可能列は、またしばらく伸ばし続けられる。これを続けていけば、かなり長い \wedge -埋め込み不可能列が作れそうであると予想が付くだろう。実際に手を動かしてみないと感覚は掴めないと思うので、試しに次を証明してみるとよい。

問題 2. $\text{tree}(3) \geq 100$ であることを証明せよ。つまり、上記ルールを遵守する初期頂点数 4 の \wedge -埋め込み不可能列で、長さ 100 以上のものが存在することを示せ。

実際には、このステップは 100 を遥かに越すことができる。それでは、一体どれくらいの長さまでこの \wedge -埋め込み不可能列の構成を続けることができるだろうか。その答えを知ると、少し驚くかもしれない。2020 年 4 月現在、Googology wiki によれば

$$\text{tree}(3) \geq 844424930131960 \approx 844 \text{ 兆}$$

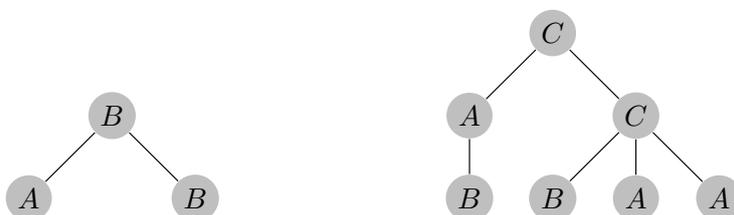
という下界が与えられているようである。なんと、うまくやれば、上記のような \wedge -埋め込み不可能列の構成は 844 兆ステップ以上続けることができるのである！ 一応、注意しておけば、本題の $\text{TREE}(3)$ はこの $\text{tree}(3)$ とは別物であり、 $\text{TREE}(3)$ の方が比べ物にならないほど巨大である。

さて、 tree 関数が爆発的に膨れ上がると聞いて、指数関数みたいなものかな、と予想する人もいるかもしれない。しかし、実は、この tree 関数は、指数関数だとか超指数関数だとかいうような

甘っちょろい代物ではないのである（その程度の代物が、数学に現れる史上最大の数に関するギネス記録を超えるはずがない！）。そういうわけで、tree 関数の増大度についても語りたいところであるが、それよりも前に本題の TREE 関数を導入することにしよう。

1.4. ラベル付き木

本題の TREE は、ただの木ではなく、ラベル付き木と呼ばれるものに関する数である。\$L\$ を集合とする。\$L\$-ラベル付き木 (\$L\$-labeled tree) とは、木 \$T\$ と関数 \$l_T: T \to L\$ の対 \$(T, l_T)\$ である。たとえば、\$l_T(\sigma) = A\$ であるとは、ノード \$\sigma\$ に \$A\$ というラベルが付いていることを意味する。図的に書くと、以下のようにノードに必ず何らかのラベルが付いているようなものである。



今回はラベルを丸で括ったが、四角で括ることもある。左のラベル付き木を \$S\$ とし、右のラベル付き木を \$T\$ としよう。筆者の好む記法を用いれば、\$S\$ と \$T\$ は以下のように表記することができる。

$$S = B * (A + B), \quad T = C * \left((A * B) + (C * (B + A + A)) \right).$$

これはラベル付き木の項表示というものであり、クラスカルの木の本定理の証明を扱う際に便利である（第 3.2 節を参照）。ただし、これはゲーゴロジストが用いている記法とは異なることに注意する。ゲーゴロジストの記法では、ラベル毎に括弧記号を割り当てる。たとえば、ラベル \$A, B, C\$ にそれぞれ括弧 \$(), [], \{\}\$ を割り当てれば、\$S\$ と \$T\$ は以下のように表される。

$$S = [(), []], \quad T = \{([], \{\{\}, (), ()\})\}.$$

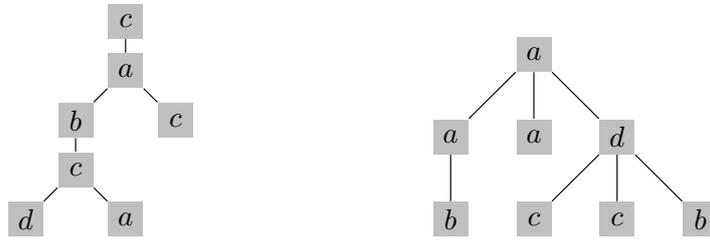
ゲーゴロジストの記法は、コンパクトであるため、具体例を弄る際には便利である。一方で、記法に一般性がないため厳密な数学的証明には適さないという欠点はあるかもしれない。どちらの記法も一長一短であるから、目的に応じて使い分けるとよい。

ラベル付き木の項表示について、もう少し詳細を説明すると、葉から順に帰納的に以下のようにして項を構成していく。

$$A * (t_1 + t_2 + t_3 + \cdots + t_n) =$$

たとえば、第 1.4 節の最初で述べた木 \$T\$ について、まずは項 \$t_1 = A * B\$ と \$t_2 = C * (B + A + A)\$ を構成して、次に項 \$T = C * (t_1 + t_2)\$ を得たと考えると分かりやすい。

問題 3. 以下の 2 つのラベル付き木の項表示を与えよ .



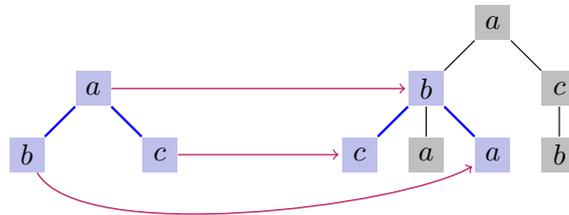
ラベル付き木の \wedge -埋め込み: 木の \wedge -埋め込み (下半束埋め込み) は, 順序 \sqsubseteq と下限 \wedge を保つ単射であった . ラベル付き木の \wedge -埋め込みは, さらにラベルを保つことを要求する . つまり, 以下のように定義する .

定義 1.8. $S = (S, \ell_S)$ と $T = (T, \ell_T)$ をラベル付き木としたとき, S から T への \wedge -埋め込み (*inf-embedding*) とは, 木の \wedge -埋め込み $h: S \rightarrow T$ であって, 次の条件を満たすものである .

$$(\forall \sigma \in S) \quad h(\ell_S(\sigma)) = \ell_T(h(\sigma)).$$

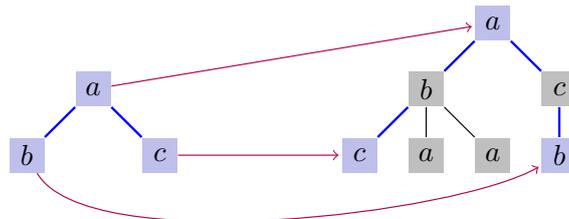
S から T への \wedge -埋め込みが存在するとき, $S \leq T$ と書く .

例 1.9. 次は木の \wedge -埋め込みにはなっているが, ラベル付き木の \wedge -埋め込みにはなっていない例である .



上の埋め込みを分析すると, 順序 \sqsubseteq と下限 \wedge は保っているが, ラベルは保っていない . 具体的には, ラベル a のノードをラベル b のノードに, ラベル b のノードをラベル a のノードに移してしまっている .

例 1.10. 次はラベル付き木の \wedge -埋め込みの例である . つまり, 順序 \sqsubseteq , 下限 \wedge , ラベルをすべて保っている .



項表示を用いれば, $a * (b + c) \leq a * (b * (c + a + a) + c * b)$ と書くことができる .

問題 4. 以下の各不等式が成立するかしないかについて答え，その理由について説明せよ．

1. $b * (a + a) \leq b * a * (a + a)$.
2. $b * (a + a * b) \leq b * (a * (a + b) + b * a)$.
3. $c * (a + b + c) \leq a * (a * (a + b + c) + b * (a + b + c) + c * (c + c + c))$.
4. $a * (b * (c + d)) \leq c * a * (b * (b * (a + a) + b * (c + a) + d * c) + d)$.

これで， $TREE(n)$ を定義する準備は整った．以後， $\{0, 1, \dots, n-1\}$ -ラベル付き木のことを n -ラベル付き木と呼ぶことにする．クラスカルの木の本定理（定理 1.6）は n -ラベル付き木についても成立する．

$TREE(n)$ とは，各 T_i の頂点数が高々 i であるような最長の n -ラベル付き木の \wedge -埋め込み不可能列 T_1, T_2, \dots, T_k の長さである．

定義 1.11. $TREE(n)$ とは，次の条件を満たす n -ラベル付き木の列 T_1, T_2, \dots, T_ℓ が存在するような最大の ℓ を表す．

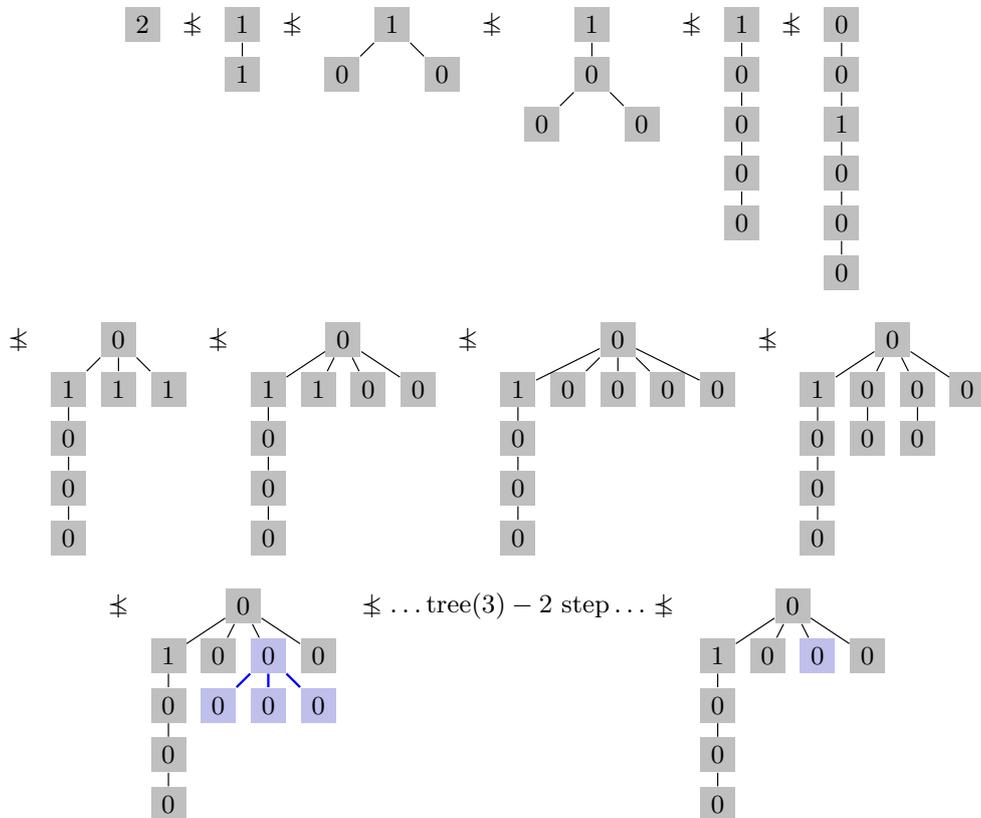
1. 任意の $1 \leq j < k \leq \ell$ について， $T_j \not\leq T_k$ である．
2. 任意の $1 \leq i \leq \ell$ について， T_i の頂点数は高々 i である．

ここで， $tree$ とは異なり， T_i 頂点数は $n+i$ ではなく i であることに注意する．したがって， $TREE$ の場合，初手は常に頂点数 1 である． $tree$ と同様に， $TREE$ は $n = 0, 1, 2$ のときには小さい．まず， $n = 0$ のとき，0-ラベル付き木は存在しないので， $TREE(0) = 0$ である． $n = 1$ のときには，1-ラベル付き木とはただの木であると思えるので， $TREE(1) = tree(0) = 1$ が分かる． $n = 2$ の場合には， $0, 1 * 1, 1$ または $1, 0 * 0, 0$ が最長の \wedge -埋め込み不可能列であり， $TREE(2) = 3$ である．



これまでと同様に，図中の \wedge -埋め込み不可能性の記号 $\not\leq$ は左右両隣だけでなく，列の中の左右両側すべてに掛かっている．このように，初手の頂点数 1 というのがなかなかネックであり， $TREE$ はあまり大きな値を取らないように見える……と思いきや， $TREE(3)$ で爆発する．その片

鱗を少しだけお見せしよう．以下が，3-ラベル付き木の \wedge -埋め込み不可能列の例である．



ここで， T_{12} から始まる $\text{tree}(3) - 2$ ステップの詳細について説明する．1-ラベル付き木がただの木に他ならないことを利用すると，青色でハイライトした部分項 $0 * (0 + 0 + 0)$ は頂点数 4 の木だと思える．このとき， $\text{tree}(3)$ の定義から，頂点数 4 の木 $0 * (0 + 0 + 0)$ を初手とする長さ $\text{tree}(3)$ の \wedge -埋め込み不可能列が存在する．この埋め込み不可能列の最後の手が頂点数 1 の木であることは明らかである．よって，初手 $0 * (0 + 0 + 0)$ かつ最終手 0 となり，この中間には $\text{tree}(3) - 2$ ステップがあるはずである．

一応， T_{12} 以降のステップの中に T_{11} 以前の木を \wedge -埋め込みできないことを確認する必要はある．しかし， T_{12} 以降のステップで変化するのは青色ハイライト部分だけであり，しかもラベルは 0 しか用いないということを利用すれば， \wedge -埋め込み不可能性は容易に示せる．

以上より， \wedge -埋め込み不可能列は $\text{tree}(3)$ ステップ以上に伸びているから，この時点で既に

$$\text{TREE}(3) > \text{tree}(3) > 844 \text{ 兆}$$

は導かれている．しかし， \wedge -埋め込み不可能列はまだまだ伸ばすことができる．まず，上図の右下の木は $T_{\text{tree}(3)+10}$ である．したがって，次の手 $T_{\text{tree}(3)+11}$ として，頂点数 $\text{tree}(3) + 11$ の木を出

このように「足す」という演算を知った人類は、 $x + x + x$ や $x + x + x + x + x$ のように x を何度も足す、という演算が有用であることに次第に気づき始める。これを簡潔に表すために、人類は「掛ける」という演算を次のように定義した。

$$x \cdot y = \underbrace{x + x + \dots + x + x}_{y \text{ 個}}$$

そして、「掛ける」という演算を知った人類は、 $x \cdot x \cdot x$ のように x を何度も掛ける、という演算の有用性に気づく。そして「累乗」という演算を次のように定義した。

$$x^y = \underbrace{x \cdot x \cdot \dots \cdot x \cdot x}_{y \text{ 個}}$$

すると、自然に x^{x^x} や $x^{x^{x^x}}$ のようなものを考える人も現れる。上方向にたくさん添字が付くのは見づらいので、 x^y のことを今後は $x \uparrow y$ と書くこととしよう。たとえば、 x^{x^x} は $x \uparrow (x \uparrow x)$ であり、 $x^{x^{x^x}}$ は $x \uparrow (x \uparrow (x \uparrow x))$ である。また、以下、括弧は省略し、これらの演算は右から順に適用するものとする。さて、累乗でも飽き足りない一部の人類は、「テトレーション^{*2}」という演算を編み出した。

$$x \uparrow\uparrow y = \underbrace{x \uparrow x \uparrow \dots \uparrow x \uparrow x}_{y \text{ 個}}$$

飽くなき人類は、更なる演算「ペンテーション」を定義する。

$$x \uparrow\uparrow\uparrow y = \underbrace{x \uparrow\uparrow x \uparrow\uparrow \dots \uparrow\uparrow x \uparrow\uparrow x}_{y \text{ 個}}$$

より一般に、クヌースの矢印記法 (*Knuth's arrow notation*) というものは以下によって定義される。

$$x \uparrow^{n+1} y = \underbrace{x \uparrow^n x \uparrow^n \dots \uparrow^n x \uparrow^n x}_{y \text{ 個}}$$

指数関数 \uparrow の段階で、指数爆発と呼ばれるほどに急増大する。そして、矢印の階層を1段駆け上がるだけで、その増大速度は急加速する。歴史的には、1925年にヒルベルト (David Hilbert) が再帰法概念を解説するために例示したものが、この演算の階層の初出であると思われる。さて、これで、グラハム数を定義する準備は整った。

定義 2.1. 次のように関数 $G(n)$ を再帰的に定義する。

$$G(0) = 4, \quad G(n+1) = 3 \uparrow^{G(n)} 3.$$

このとき、 $G(64)$ のことをグラハム数 (*Graham's number*) と呼ぶ。

^{*2} テトレーションを「超冪」と訳す(これは super-exponentiation の訳なのだろうか?) という人が極少数いるようだが、その訳語は決して勧めない(ネット起源の訳語だろうか? テトレーションを「超冪」と訳している専門文献は見ることがないと思う。「超指数」と訳すならばまだ良いと思うが)。「超冪」という言葉は、遥かに数学的に重要な別概念である ultrapower の訳語として用いられている。

グラハム数は、あくまでラムゼー理論のある未解決問題の上界として与えられたものであり、いかなる意味でも、数学的に本質的な数ではない。したがって、たとえばなぜ 64 なのか、などといった意味を考えても徒勞に終わる。

クヌースの矢印表記の階層（ヒルベルトの階層）は定義が単純明快であるが、少し分析をしづらい。このため、クヌースの矢印表記よりは分析が容易な、急増加階層 (*fast-growing hierarchy*) と呼ばれる自然数上の関数の階層を導入しよう。その有限段階は、以下によって定義される。

$$\begin{aligned} f_0(x) &= x + 1, \\ f_{n+1}(x) &= \underbrace{f_n \circ f_n \circ \cdots \circ f_n}_x(x), \\ f_\omega(x) &= f_x(x). \end{aligned}$$

具体的に、急増加階層の最初のステップを記述してみると、

$$f_1(x) = 2x, \quad f_2(x) = 2^x x \geq 2 \uparrow x$$

であることが分かる。一般に、急増加階層とクヌースの矢印表記を比較してみると、以下を得る。

$$f_{n+2}(x) \geq \underbrace{2 \uparrow^n 2 \uparrow^n \cdots \uparrow^n 2 \uparrow^n}_x x \geq 2 \uparrow^{n+1} x.$$

問題 6. 任意の自然数 $x \geq 3$ について、 $2 \uparrow^{n+1} x \geq 3 \uparrow^n 3$ となることを示せ。

急増加関数の階層の発想は、さらに先のステップに進めることができる。たとえば、各自然数 $n \in \mathbb{N}$ について、以下のような関数 $f_{\omega+n}: \mathbb{N} \rightarrow \mathbb{N}$ を定義できることは明らかであろう。

$$\begin{aligned} f_{\omega+n+1}(x) &= \underbrace{f_{\omega+n} \circ f_{\omega+n} \circ \cdots \circ f_{\omega+n}}_x(x), \\ f_{\omega+\omega}(x) &= f_{\omega+x}(x). \end{aligned}$$

記号 ω は何なのかと思うかもしれないが、現段階では、ただの形式的な記号列だと思ってもらっていい。重要なことは、上で定義した f_α が自然数上の関数であること、つまり有限の値を入力すれば有限の値を出力するということである。

それでは、急増加関数を用いて、グラハム数の上界を与えよう。

命題 2.2. $G(64) < f_{\omega+1}(64)$.

Proof. まず、 $f_{\omega+1}(64)$ は、以下のような数列から得ることができる。

$$F(0) = 64, \quad F(n+1) = f_\omega(F(n)).$$

ここで、 $f_{\omega+1}(64) = F(64)$ である。帰納的に $F(n) \geq G(n) + 2$ であることを示す。まず、 $F(0) \geq G(0) + 2$ は成立している。つづいて、上で述べた、急増加関数とクヌースの矢印記法の比

較および帰納的仮定を用いれば,

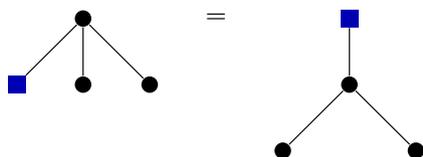
$$F(n+1) = f_\omega(F(n)) \geq f_\omega(G(n) + 2) \geq 2^{G(n)+1}x \geq 3 \uparrow^{G(n)} 3 + 2 = G(n+1) + 2$$

を得る. よって主張は示された. □

2.2. 関数 tree の増大度

グラハム数の定義は終わったので, それでは話を TREE(3) に戻そう. TREE(3) の下界を与えるためには, 関数 tree(x) の分析をすればよいのであった. したがって, ふたたび木の話に入ることしよう.

さて, われわれが扱う木は根付き木であり, これまでは根は最上位に記述していた. しかし, 今後は, 都合上, 木を傾けたりしたいので, 根を最上位に配置する代わりに, 根を青い四角で表すことにする.



上図の左側が新しい記法での根付き木の表記であり, 右側がこれまでのように根を最上位に配置した木である. 根が指定されていれば, 根をつまんで持ち上げることによって, われわれはいつでも根を最上位に配置し直すことができる.

いま, 自然数の n 組 $\bar{a} = (a_{n-1}, a_{n-2}, \dots, a_1, a_0)$ で $a_{n-1} > 0$ であるものが与えられているとしよう. このとき, 次の木 $T_{\bar{a}}$ を考える.



括弧内の値は, 省略されている頂点数を表す. 木 $T_{\bar{a}}$ の頂点数は

$$a_{n-1} + a_{n-2} + \dots + a_1 + a_0 + 2n - 2$$

であることに注意する. たとえば, (3, 2) および (1, 0, 1, 0) に対応する木はそれぞれ以下である.



ここで, 数列の値をコードしている部分を紫色でハイライトしている. もう少し分かりやすい例を挙げておけば, (1, 3, 2, 0, 1, 2) に対応する木は以下である.



上の説明では少し分かりにくいだが, 長さ 1 の列 (a) の場合には, $(a - 1)$ 個の頂点の他に左右両端の頂点を含むので, 頂点数 $a + 1$ の直線状の木となる. また, 例外として列 (0) を認めることに

すると, $T_{(0)}$ は頂点数 1 の木である. 数列 \bar{a} の最初の項目が正でない場合には, 最初に正の数が見れるまで飛ばして読むことにする. たとえば, $T_{(0,3,1)} = T_{(3,1)}$ および $T_{(0,0,0,2,2)} = T_{(2,2)}$ である.

いま, tree 関数の定義において, 利用してよい木の種類を大幅に制限することにしよう. まずは, 長さ $k+1$ 以下の数列 \bar{a} に関する $T_{\bar{a}}$ のみを利用してよい, という制限を与えよう. このような限られた木だけを用いた場合の初期頂点数 $n+1$ の \wedge -埋め込み不可能列の最大長を $\text{tree}_k(n)$ と表す.

定義 2.3. 次の条件を満たす木の列 T_1, T_2, \dots, T_ℓ が存在するような最大の ℓ を $\text{tree}_k(n)$ と表す.

1. 任意の $1 \leq i < j \leq \ell$ について, $T_i \not\leq T_j$ である.
2. 任意の $1 \leq i \leq \ell$ について, T_i の頂点数は高々 $n+i$ である.
3. 任意の $1 \leq i \leq \ell$ について, 長さ $k+1$ 以下の数列 \bar{a} が存在して, $T_i = T_{\bar{a}}$ である.

使える木が少ないのだから, 明らかに

$$j \leq k \implies \text{tree}_j(n) \leq \text{tree}_k(n) \leq \text{tree}(n)$$

である. 実際, $\text{tree}_k(n)$ は $\text{tree}(n)$ とは比べ物にならないほど遥かに小さい. しかし, それでも, グラハム数という「ギネス記録」程度には差し迫る.

この tree_k 関数の分析のために便利な道具が辞書式順序である.

定義 2.4. 自然数の m 組全体の集合 \mathbb{N}^m 上の辞書式順序 (*lexicographical order*) とは, 以下によって定義される順序である: 自然数の m 組 $\bar{x} = (x_1, x_2, \dots, x_m)$ および $\bar{y} = (y_1, y_2, \dots, y_m)$ について,

$$\bar{x} <_{\text{lex}} \bar{y} \iff (\exists i \leq m)(\forall k < i) [x_k = y_k \wedge x_i < y_i]$$

とし, $\bar{x} <_{\text{lex}} \bar{y}$ または $\bar{x} = \bar{y}$ であるとき, $\bar{x} \leq_{\text{lex}} \bar{y}$ と書く.

辞書式順序という名の由来は, 辞書における語句の記載順のように, 現れる文字を先頭から順に見ていき, 最初に違いが現れたところで順序比較をするためである. つまり, 「あいうえお順」に $0, 1, 2, \dots$ と数字を割り当てていくと, 通常の辞書に語句が現れる順序は, 上の辞書式順序で定義されるものと全く同一である.

\mathbb{N}^m 上の辞書式順序の順序型は, ω^m と表記される. この辞書式順序に注目しているときは, 各 $(a_{m-1}, a_{m-2}, \dots, a_1, a_0) \in \mathbb{N}^m$ を形式的に

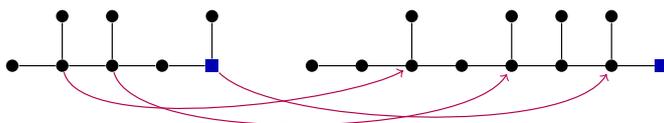
$$\omega^{m-1} \cdot a_{m-1} + \omega^{m-2} \cdot a_{m-2} + \dots + \omega \cdot a_1 + a_0$$

と表し, しばしば \leq_{lex} を単に \leq と書くことがある.

補題 2.5. 長さ m の任意の自然数列 \bar{u}, \bar{v} について, 以下が成立する .

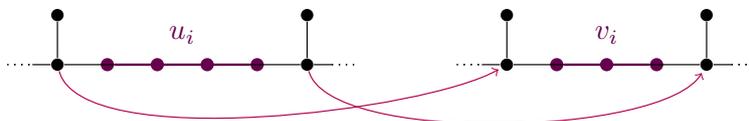
$$\bar{v} <_{\text{lex}} \bar{u} \implies T_{\bar{u}} \not\preceq T_{\bar{v}}.$$

Proof. $T_{\bar{u}}$ から $T_{\bar{v}}$ への \wedge -埋め込みが与えられたと仮定する . これが下限 \wedge を保つためには , 分岐ノードを分岐ノードに移さなければならない .



もし $\bar{v} = (v_1, \dots, v_m)$ の最初の項目 v_1 が 0 であって , $\bar{u} = (u_1, \dots, u_m)$ の最初の項目 u_1 が正ならば , 木 $T_{\bar{v}}$ の方が木 $T_{\bar{u}}$ より分岐ノードの個数が少ないので , \wedge -埋め込みが存在しないことがわかる . したがって , 帰納的に , \bar{u} と \bar{v} の最初の項目が共に正であると仮定できる .

\bar{u} と \bar{v} の最初の項目が共に正である場合には , 木 $T_{\bar{u}}$ と木 $T_{\bar{v}}$ の分岐ノードの数は等しい . 仮定より $\bar{v} <_{\text{lex}} \bar{u}$ なので , $v_i < u_i$ となる最初の i を取る . \wedge -埋め込みは , $T_{\bar{u}}$ の分岐ノードを木 $T_{\bar{v}}$ の分岐ノードに順番通りに移す . また , \wedge -埋め込みは順序を保つことから , $T_{\bar{u}}$ におけるこの中間部は , $T_{\bar{v}}$ における中間部に移される .



この中間部において , $T_{\bar{u}}$ の頂点数は u_i であり , $T_{\bar{v}}$ の頂点数は v_i である . しかし , \wedge -埋め込みは単射であるから , $u_i > v_i$ より , これは不可能である . $i = 1, m$ の場合には , コード部が分岐ノードに囲まれていないが , 議論は同様である . \square

一般に , 木の \wedge -埋め込み可能性のチェックは容易ではないが , 数列の辞書式順序での比較は容易である . 補題 2.5 の便利なところは , 辞書式順序で手前に来るものをどんどん選んでいくことにより , それが \wedge -埋め込み不可能列となっていることが保証されるということである .

以下 , $\bar{a} = (a_1, \dots, a_k)$ に対する木 $T_{\bar{a}}$ を $[a_1, \dots, a_k]$ と書くことにする . それでは , tree_k 関数の増大度の評価に入ろう .

命題 2.6. 任意の自然数 k, n について , 以下の不等式が成立する .

$$\text{tree}_1(2n + 1) > n^2, \quad \text{tree}_{k+1}(2n + 1) \geq \underbrace{\text{tree}_k \circ \text{tree}_k \circ \dots \circ \text{tree}_k}_n(n)$$

証明. まず, $\text{tree}_0(n)$ を求める. 初手の頂点数は $n + 1$ であり, ルールを遵守する \wedge -埋め込み不可能列のうちでは, 明らかに

$$[n] \rightarrow [n - 1] \rightarrow [n - 2] \rightarrow \cdots \rightarrow [1] \rightarrow [0]$$

が最大長のものであるから, その長さは $\text{tree}_0(n) = n + 1$ である.

つづいて, tree_{k+1} について考察する. $\text{tree}_k(n)$ の値を実現する \wedge -埋め込み不可能列の初手を $\bar{a} = [a_1, \dots, a_k]$ とする. 定義より, この頂点数は $n + 1$ である.

$$\bar{a} = [a_1, \dots, a_k] \rightarrow \cdots \rightarrow [c_i, \dots, c_k] \rightarrow \cdots \rightarrow [0].$$

$\underbrace{\hspace{15em}}_{\text{tree}_k(n) \text{ ステップ}}$

まず, 木 $[n, \bar{a}] := [n, a_1, \dots, a_k]$ の頂点数が $2n + 2$ であることを容易に確認できる. それでは, $\text{tree}_{k+1}(2n + 1)$ の下限を求めてみよう. 木 $[n, \bar{a}]$ の頂点数は $2n + 2$ であったから, 初手として $T_1 = [n, \bar{a}]$ を取ることができる. そして, \bar{a} の選択より, $\text{tree}_k(n)$ ステップ掛けて, $[n, \bar{a}]$ から $[n, 0, 0, \dots, 0]$ に辿り着く.

$$[n, \bar{a}] \rightarrow \cdots \rightarrow [n, \overbrace{0, \dots, 0}^{(i-1) \text{ 個}}, c_i, \dots, c_k] \rightarrow \cdots \rightarrow [n, \overbrace{0, 0, \dots, 0}^k].$$

$\underbrace{\hspace{15em}}_{\text{tree}_k(n) \text{ ステップ}}$

次に, $\text{tree}_k(\text{tree}_k(n))$ の値を実現する \wedge -埋め込み不可能列の初手を \bar{a}' とする.

$$\bar{a}' \rightarrow \cdots \rightarrow [c'_j, \dots, c'_k] \rightarrow \cdots \rightarrow [0].$$

$\underbrace{\hspace{15em}}_{\text{tree}_k(\text{tree}_k(n)) \text{ ステップ}}$

先ほどと同様にして, 木 \bar{a}' の頂点数は $\text{tree}_k(n) + 1$ であり, 木 $[n - 1, \bar{a}']$ の頂点数は $\text{tree}_k(n) + n + 1$ である. また, 任意の \bar{v} について, $(n, \bar{v}) >_{\text{lex}} (n - 1, \bar{a})$ であるから, 補題 2.5 より, $[n, \bar{v}] \not\leq [n - 1, \bar{a}]$ である. したがって, 木 $[n - 1, \bar{a}']$ を第 $\text{tree}_k(n) + 2$ 手で打つことができる. そして, \bar{a}' の選択より, $\text{tree}_k(\text{tree}_k(n))$ ステップ掛けて, $[n - 1, \bar{a}']$ から $[n - 1, 0, 0, \dots, 0]$ に辿り着く.

$$[n, \overbrace{0, 0, \dots, 0}^k] \rightarrow [n - 1, \bar{a}'] \rightarrow \cdots \rightarrow [n - 1, \overbrace{0, \dots, 0}^{(j-1) \text{ 個}}, c'_j, \dots, c'_k] \rightarrow \cdots \rightarrow [n - 1, \overbrace{0, 0, \dots, 0}^k].$$

$\underbrace{\hspace{15em}}_{\text{tree}_k(\text{tree}_k(n)) \text{ ステップ}}$

これを繰り返すことにより, 非常に粗い評価であるが, 以下を得ることができる.

$$\text{tree}_{k+1}(2n + 1) > \sum_{i=1}^n \underbrace{\text{tree}_k \circ \text{tree}_k \circ \cdots \circ \text{tree}_k}_{i \text{ 個}}(n)$$

特に, $\text{tree}_0(n) = n$ と上の不等式を合わせれば, 粗い評価であるが, $\text{tree}_1(2n + 1) > n^2$ も得る. □

以下は粗い評価であるが，急増加階層との比較をしておくとも便利である．

命題 2.7. 任意の自然数 k および $n \geq 1$ について，以下が成立する．

$$\text{tree}_k(2^k n) \geq \mathbf{f}_k(n).$$

Proof. $s \leq 2$ について，主張は明らかに成立する．いま，漸化式 $c_2(n) = 2n + 1$ かつ $c_{k+1}(n) = 2c_k(n) + 1$ を解くと，以下を得る．

$$c_k(n) = 2^{k-1}n + 2^{k-1} - 1.$$

帰納的に $\text{tree}_k(c_k(n)) \geq \mathbf{f}_k(n)$ を示そう．命題 2.6 より， $\text{tree}_1(n) \geq n^2 + 1 \geq 2n = \mathbf{f}_1(n)$ も成立する．したがって，命題 2.6 と急増加階層の定義より， $\text{tree}_2(2n + 1) \geq \mathbf{f}_2(n)$ が成り立つ．

つづいて， k について主張が成立すると帰納的に仮定する．もし $n \geq k$ ならば，

$$c_k(n) = 2^{k-1}n + 2^{k-1} - 1 \leq 2^n \cdot n = \mathbf{f}_2(n) < \text{tree}_2(2n + 1) \leq \text{tree}_2(\text{tree}_2(n)) \leq \text{tree}_3(n)$$

が成立する．帰納的仮定より，以下の不等式を得る．

$$\mathbf{f}_k(\mathbf{f}_k(n)) \leq \text{tree}_k(c_k(\mathbf{f}_k(n))) \leq \text{tree}_k(c_k(\text{tree}_k(c_k(n)))).$$

いま， $n \geq 1$ ならば $c_k(n) \geq k$ なので，特に $\text{tree}_k(c_k(n)) \geq k$ である．よって，前の 2 つの不等式を組み合わせると，以下を導くことができる．

$$\mathbf{f}_k(\mathbf{f}_k(n)) \leq \text{tree}_k(\text{tree}_3(\text{tree}_k(c_k(n)))).$$

同様の議論より，一般に，以下の不等式を得る．

$$\underbrace{\mathbf{f}_k \circ \mathbf{f}_k \circ \cdots \circ \mathbf{f}_k}_{k+1 \text{ 個}}(n) \leq \underbrace{\text{tree}_k \circ \text{tree}_k \circ \cdots \circ \text{tree}_k}_{2k+1 \text{ 個}}(c_k(n)).$$

したがって，命題 2.6 と急増加階層の定義より，

$$\begin{aligned} \mathbf{f}_{k+1}(n) &= \underbrace{\mathbf{f}_k \circ \mathbf{f}_k \circ \cdots \circ \mathbf{f}_k}_{n \text{ 個}}(n) \leq \underbrace{\text{tree}_k \circ \text{tree}_k \circ \cdots \circ \text{tree}_k}_{2n \text{ 個}}(c_k(n)) \\ &\leq \text{tree}_{k+1}(2c_k(n) + 1) = \text{tree}_{k+1}(c_{k+1}(n)). \end{aligned}$$

以上より，帰納法および $n \geq 1$ について $c_k(n) \leq 2^k n$ であることを用いれば，目的の不等式が導かれる． \square

命題 2.6 と 2.7 から得られる便利な帰結として， $n > 1$ に対する以下の不等式が得られる．

$$\text{tree}_n(\text{tree}_3(n)) \geq \text{tree}_n(\text{tree}_2(2n + 1)) > \text{tree}_n(\mathbf{f}_2(n)) = \text{tree}_n(2^n n) \geq \mathbf{f}_n(n) = \mathbf{f}_\omega(n).$$

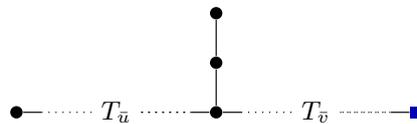
数列の対: 数列に対応する木を考えることで, 数列の長さに応じて急増加関数の階層を登ることができた. さらに上のステップに進むために, 次は数列の対に木を割り当てる. 自然数の m 組 $\bar{u} = (u_{m-1}, u_{m-2}, \dots, u_0)$ と n 組 $\bar{v} = (v_{n-1}, v_{n-2}, \dots, v_0)$ が与えられたとき, その対を,

$$(\bar{u}; \bar{v}) = (u_{m-1}, u_{m-2}, \dots, u_0; v_{n-1}, v_{n-2}, \dots, v_0)$$

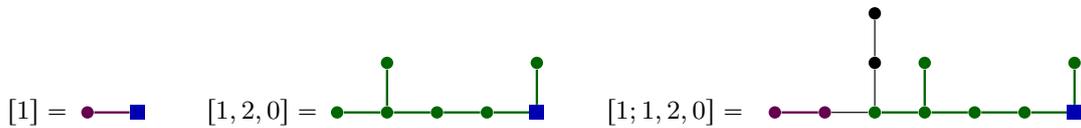
と表すことにする. 対 $(\bar{u}; \bar{v})$ のことをしばしば以下のように形式的に書く.

$$\omega^{\omega+m-1}u_{m-1} + \omega^{\omega+m-2}u_{m-2} + \dots + \omega^{\omega}u_1 + \omega^{n-1}v_{n-1} + \omega^{n-2}v_n + \dots + \omega v_1 + v_0.$$

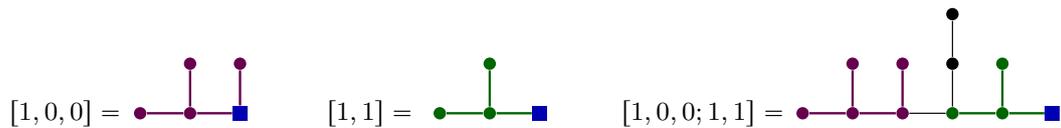
なぜわざわざこんなややこしい形式表記を考えるのかと思うかもしれないが, 実は便利なこともある. それについては, 後で説明する. さて, $u_m > 0$ であるとき, $(\bar{u}; \bar{v})$ に対する木 $[\bar{u}; \bar{v}]$ を以下のように定義しよう.



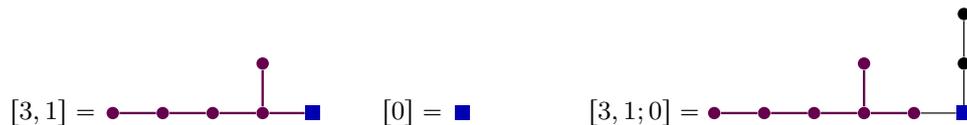
ただし, 点線部には木 $T_{\bar{u}}$ と $T_{\bar{v}}$ が省略されている. さらに, $T_{\bar{u}}$ は左端の頂点を含んでおり, $T_{\bar{v}}$ は中央の頂点と右端の根を含んでいる. 少し分かりにくいかもしれないので, 具体例を挙げよう. たとえば, $\omega^{\omega} + \omega^2 + \omega 2 = (1; 1, 2, 0)$ に対応する木を書いてみると, 以下のようなになる.



もうひとつ例を挙げておけば, たとえば $\omega^{\omega+2} + \omega + 1 = (1, 0, 0; 1, 1)$ に対応する木は以下である.



注意点として, 右側が $[0]$ であることが許される. たとえば, $\omega^{\omega+1}3 + \omega^{\omega} = (3, 1; 0)$ に対応する木は以下のように与えられる.



それでは, 今度は, $[\bar{u}; \bar{v}]$ の形の木しか利用してはいけない, という制約を加えることにしよう. ただし, \bar{u} は長さ k 以下の列であり, \bar{v} は如何なる長さでも良い, という制限を加える. この条件

の下での、 \wedge -埋め込み不可能列の最大長を与える関数を $\text{tree}_{\omega+k}$ と定義する．任意の自然数 i および $j \leq k$ について，以下が成立する．

$$\text{tree}_i(n) \leq \text{tree}_{\omega+j}(n) \leq \text{tree}_{\omega+k}(n) \leq \text{tree}(n).$$

この関数の分析のために便利な道具は，ふたたび辞書式順序である．有限列の対 $(\bar{u}_0; \bar{u}_1), (\bar{v}_0; \bar{v}_1)$ が与えられたとき，先頭に 0 を加えることによって， \bar{u}_i と \bar{v}_i は等しい長さであると仮定できる．この仮定の下で，有限列の対の辞書式順序を次によって定義する．

$$(\bar{u}_0; \bar{v}_0) <_{\text{lex}} (\bar{u}_1; \bar{v}_1) \iff \bar{u}_0 <_{\text{lex}} \bar{u}_1 \text{ or } (\bar{u}_0 = \bar{u}_1 \text{ and } \bar{v}_0 <_{\text{lex}} \bar{v}_1)$$

元の有限列の辞書式順序 $<_{\text{lex}}$ を $<$ と略記すれば，上記の定義は，長さ 2 の列に対する辞書式順序の定義と全く同じであることが分かるだろう．このアイデアを拡張すれば，有限列の対だけではなく，有限列の有限列に対する辞書式順序も容易に導入でき，上の定義は辞書式順序の入れ子の特殊なものに過ぎない．このように順序を入れ子にするという発想は順序理論においては極めて有用であり，今後も活用していく．

さて，補題 2.5 と同様の議論によって，次を示すことができる．

補題 2.8. 任意の自然数列の対 $(\bar{u}_0; \bar{u}_1), (\bar{v}_0; \bar{v}_1)$ について，以下が成立する．

$$(\bar{v}_0; \bar{v}_1) <_{\text{lex}} (\bar{u}_0; \bar{u}_1) \implies [\bar{u}_0; \bar{u}_1] \not\leq [\bar{v}_0; \bar{v}_1].$$

問題 7. 補題 2.8 を証明せよ．

次もまた粗い評価であるが，われわれの目的には十分である．

命題 2.9. 任意の自然数 n について以下が成立する．

$$\text{tree}_{\omega+1}(2n+3) \geq \mathbf{f}_{\omega+1}(n).$$

Proof. $n \leq 1$ の場合には明らかである． $n > 1$ の場合には，まず，初手で木 $[n+2; 0]$ を配置する．この木の頂点数は $n+5 \leq 2n+3$ であるから，この初手はルールを遵守している．次に， \bar{a}_0 を $\text{tree}_3(n)$ を実現する列の初手の木とすると，これは頂点数 $n+1$ である．

$$\underbrace{\bar{a}_0 \rightarrow \cdots \rightarrow [0]}_{\text{tree}_3(n) \text{ ステップ}}$$

これについて， $[n+1; \bar{a}_0]$ は頂点数 $2n+5$ であるから，第 2 手として $[n+1; \bar{a}_0]$ を選べる．ここで，補題 2.8 より， $[n+2; 0] \not\leq [n+1; \bar{a}_0]$ である．ここから $\text{tree}_3(n)$ ステップ掛けて，木を $[n+1; \bar{a}_0]$ から $[n+1; 0]$ に変化させていく．

$$[n+2; 0] \rightarrow \underbrace{[n+1; \bar{a}_0] \rightarrow \cdots \rightarrow [n+1; 0]}_{\text{tree}_3(n) \text{ ステップ}}$$

つづいて，第 $\text{tree}_3(n) + 2$ 手目の木を考えたい． \bar{u}_0 を $\text{tree}_n(\text{tree}_3(n))$ を実現する列の初手の木とする．命題 2.7 の後に説明した不等式より， $\text{tree}_n(\text{tree}_3(n)) \geq \mathbf{f}_\omega(n)$ であるから，この列の長さは $\mathbf{f}_\omega(n)$ を遥かに上回る．

$$\underbrace{\bar{u}_0 \rightarrow \cdots \rightarrow [0]}_{\geq \mathbf{f}_\omega(n) \text{ ステップ}}.$$

いま， \bar{u}_0 の頂点数は $\text{tree}_3(n) + 1$ である． $n \geq 3$ の場合には， \bar{u}_0 は数列として長さ 3 以上であるから，頂点を 2 つ付加すれば，木 $[1, \bar{u}_0]$ を得る．第 $\text{tree}_3(n) + 2$ 手目として $[n; 1, \bar{u}_0]$ を取ると，頂点数は $\text{tree}_3(n) + n + 6$ であるから，これはルールを遵守する． $n = 2$ の場合には，頂点を 3 つ付加して木 $[1, 0, \bar{u}_0]$ を得た後，同様の議論を行う．いま，列の組として $(n+1; 0) >_{\text{lex}} (n; 1, \bar{u}_0)$ であるから，補題 2.8 より， $[n+1; 0] \preceq [n; 1, \bar{u}_0]$ である．ここから $\text{tree}_n(\text{tree}_3(n)) \geq \mathbf{f}_\omega(n)$ ステップを掛けて，木を $[n; 1, \bar{u}_0]$ から $[n; 1, 0, \dots, 0]$ に変化させていく．ここで，0 の個数は \bar{u}_0 の長さと同じ．

$$[n+1; 0] \rightarrow \underbrace{[n; 1, \bar{u}_0] \rightarrow \cdots \rightarrow [n; 1, 0, 0, \dots, 0]}_{\geq \mathbf{f}_\omega(n) \text{ ステップ}}.$$

次に， \bar{a}_1 を $\text{tree}_3(\mathbf{f}_\omega(n))$ を実現する列の初手の木とすると，これは頂点数 $\mathbf{f}_\omega(n)$ である．

$$\underbrace{\bar{a}_1 \rightarrow \cdots \rightarrow [0]}_{\text{tree}_3(\mathbf{f}_\omega(n)) \text{ ステップ}}.$$

これについて， $[n; \bar{a}_1]$ は頂点数 $\mathbf{f}_\omega(n) + 4$ であるから，次の手として $[n; \bar{a}_1]$ を選べる． $(1, \bar{u}_0)$ と $(1, 0, \dots, 0)$ は同じ長さ 4 以上の数列であり， \bar{a}_1 は長さ 3 以下であるから， $(n; 1, 0, \dots, 0) >_{\text{lex}} (n; \bar{a}_1)$ であり，補題 2.8 より， $[n; 1, 0, \dots, 0] >_{\text{lex}} [n; \bar{a}_1]$ を得る．ここから $\text{tree}_3(\mathbf{f}_\omega(n))$ ステップ掛けて，木を $[n; \bar{a}_1]$ から $[n; 0]$ に変化させていく．

$$[n; 1, 0, 0, \dots, 0] \rightarrow \underbrace{[n; \bar{a}_1] \rightarrow \cdots \rightarrow [n; 0]}_{\text{tree}_3(\mathbf{f}_\omega(n)) \text{ ステップ}}.$$

次に， $n' = \mathbf{f}_\omega(n)$ について， \bar{u}_1 を $\text{tree}_{n'}(\text{tree}_3(n'))$ を実現する列の初手の木とすれば，命題 2.7 の後に説明した不等式より， $\text{tree}_{n'}(\text{tree}_3(n')) \geq \mathbf{f}_\omega(n') = \mathbf{f}_\omega(\mathbf{f}_\omega(n))$ であるから，この列の長さは $\mathbf{f}_\omega(\mathbf{f}_\omega(n))$ を遥かに上回る．

$$\underbrace{\bar{u}_1 \rightarrow \cdots \rightarrow [0]}_{\geq \mathbf{f}_\omega(\mathbf{f}_\omega(n)) \text{ ステップ}}.$$

このとき， \bar{u}_1 の頂点数は $\text{tree}_3(n') = \text{tree}_3(\mathbf{f}_\omega(n))$ であり，頂点を 2 つか 3 つ付加して，木 $[1, \bar{u}_1]$ を得る．次の手として， $[n-1; 1, \bar{u}_1]$ を取ると，頂点数は $c = 4, 5$ について $\text{tree}_3(\mathbf{f}_\omega(n)) + n + c$ であり，これはルールを遵守する．ここから $\text{tree}_{n'}(\text{tree}_3(\mathbf{f}_\omega(n))) > \mathbf{f}_\omega(\mathbf{f}_\omega(n))$ ステップを掛けて，木を $[n-1; 1, \bar{u}_1]$ から $[n-1; 1, 0, \dots, 0]$ に変化させていく．

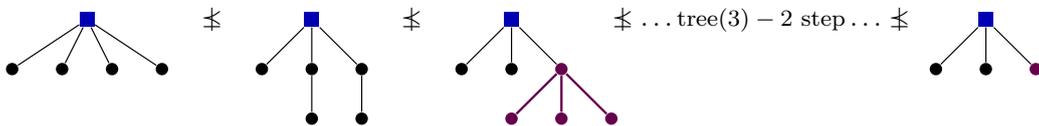
$$[n; 0] \rightarrow \underbrace{[n-1; 1, \bar{u}_1] \rightarrow \cdots \rightarrow [n-1; 1, 0, 0, \dots, 0]}_{\geq \mathbf{f}_\omega(\mathbf{f}_\omega(n)) \text{ ステップ}}.$$

このように、ルールを遵守したまま、 f_ω をどんどん累積していくことができる。これは最終的に f_ω の n 回の累積を行うので、 \wedge -埋め込み不可能列の構成は、 $f_{\omega+1}(n) = f_\omega^n(n)$ ステップ以上続けることができる。よって、目的の不等式 $\text{tree}_{\omega+1}(2n+3) \geq f_{\omega+1}(n)$ が示された。□

定理 2.10. TREE(3) はグラハム数よりも大きい。実際、 $\text{tree}(4)$ の時点でグラハム数より大きい。

$$\text{TREE}(3) > \text{tree}(4) > G(64).$$

Proof. $\text{tree}(4)$ の場合、初手は頂点数 5 以下の木を選ぶのであった。したがって、初期頂点数 5 の以下の \wedge -埋め込み不可能列 T_1, T_2, \dots として、まずは以下のものを考える。



ここで、第 3 手から始まる $\text{tree}(3) - 2$ ステップの詳細について説明する。紫色でハイライトした部分木は頂点数 4 の木だと思える。このとき、 $\text{tree}(3)$ の定義から、この頂点数 4 の木を初手とする長さ $\text{tree}(3)$ の \wedge -埋め込み不可能列 S_1, S_2, \dots が存在する。この埋め込み不可能列の最後の手が頂点数 1 の木であることは明らかである。よって、この中間には $\text{tree}(3) - 2$ ステップがあるはずである。また、 S_1 は 3 分岐以上のノードを持つ任意の木に \wedge -埋め込み可能であるから、 S_2 以降は高々 2 分岐ノードを持たない。

上図の木の列が \wedge -埋め込み不可能列となっていることは容易に確認できるが、ここではしっかりと証明を与えよう。まず、上図の初手 T_1 は 4 分岐木であり、 T_2 以降は高々 3 分岐ノードしか持たないので、 T_2 以降への下限を保つ埋め込みは存在しない。次に、 T_2 の根の 3 分岐のうち 2 つの先が頂点数 2 である。まず、 T_3 については 3 分岐ノードを 2 つ持つが、どちらについても、3 分岐の先で頂点数が 2 以上になるものは高々 1 つであるから、 T_2 の T_3 への \wedge -埋め込みは存在しない。 T_4 以降は紫部分が S_2, S_3, \dots と変化していくが、これらは高々 2 分岐ノードしか持たないので、 T_4 以降では、根のみが 3 分岐以上のノードである。しかし、 T_4 以降について、根の 3 分岐の先で頂点数が 2 以上になるものは高々 1 つであるから、 T_2 の T_4 への \wedge -埋め込みは存在しない。残りについては、 S_1, S_2, \dots が \wedge -埋め込み不可能列であったことから従う。

ここまで描いた木は、いずれも根が 3 分岐である。したがって、これらの木は、3 分岐ノードを持つ木にしか \wedge -埋め込みできない。よって、高々 2 分岐ノードしか持たない木の任意の \wedge -埋め込み不可能列をここから先に加えても、 \wedge -埋め込み不可能列であることは保たれる。さて、次の手は第 $\text{tree}(3) + 3$ 手目であるから、頂点数 $\text{tree}(3) + 7$ の木が許容される。

さて、 $\text{tree}_{\omega+k}$ の定義で許容される木は高々 2 分岐ノードしか持たない。したがって、たとえば $\text{tree}_{\omega+1}(\text{tree}(3) + 6)$ の値を実現する \wedge -埋め込み不可能列は、初期頂点数 $\text{tree}(3) + 7$ であるから、上で図示した \wedge -埋め込み不可能列の後に加えることができる。つまり、上図の後に長さ

$\text{tree}_{\omega+1}(\text{tree}(3) + 6)$ の列が付加されるのであるから、少なくとも

$$\text{tree}(4) > \text{tree}(3) + 2 + \text{tree}_{\omega+1}(\text{tree}(3) + 6)$$

という下界を与えることができる。前節の $\text{TREE}(3)$ と $\text{tree}(3)$ の下限の計算および命題 2.2 と 2.9 を上の不等式と組み合わせると、

$$\text{TREE}(3) > \text{tree}(4) > \text{tree}_{\omega+1}(\text{tree}(3)) > \text{tree}_{\omega+1}(131) \geq f_{\omega+1}(64) \geq G(64)$$

が成立するから、目的の不等式を得る。□

2.3. 素朴順序数論

前節では、急増加階層と呼ばれる、関数の階層を取り扱った。

$$f_0, f_1, f_2, f_3, \dots, f_n, f_{n+1}, \dots, f_\omega, f_{\omega+1}, \dots, f_{\omega+n}, f_{\omega+n+1}, \dots$$

右にひとつ進むだけで爆発的に増大度が上がる。たとえば、 f_2 は指数関数レベル、 f_3 はテトラエーションのレベルであり、 $f_{\omega+1}$ はグラハム数を定義する関数 G のレベルであった。そして、 tree や TREE レベルの増大度の関数はそれらを遥かに凌駕すると予想できる。このような急増加関数の増大度を測るのに良い指標が、順序数 (*ordinal*) と呼ばれるものである。より正確には、順序数表記と呼ばれるものである。

順序数の概念は、解析学の一分野である三角級数論の研究において、例外点の“個数”を数える道具として、1870 年頃にカントールが考案した。ここで、例外点の個数が有限であれば何も問題はないのだが、例外点が無限個ある場合もあり得る。そして、例外点が「十分に少ない」のであれば、三角級数に関するある性質が成り立つのであるが、ここで無限個の中でも、「十分少ない無限個」と「多い無限個」がある。このようなものを取り扱うために、数のカウントを有限から無限へと拡張したものが、順序数の起源である。

それでは、順序数の理論の応用に入ろう。たとえば、上で書いた急増加階層の添字部分は、順序数の起源的な発想に基づき、カウントを有限から無限へと拡張したものと考えることもできる。

$$0, 1, 2, 3, \dots, n, n+1, \dots, \omega, \omega+1, \dots, \omega+n, \omega+n+1, \dots$$

しかし、順序数を《数の無限への拡張》と考えるだけでは、順序数の一側面しか捉えていない。むしろ、数学的定義の上では、順序数 (*ordinal*) とは、ある種の《有限性》を持つ順序構造 (の順序型) である。数学における順序の概念を説明するために、2 項関係 \leq に関する次の 3 つの規則を考えよう。

$$\frac{}{x \leq x} \text{ (反射律)} \quad \frac{x \leq y \quad y \leq z}{x \leq z} \text{ (推移律)} \quad \frac{x \leq y \quad y \leq x}{x = y} \text{ (反対称律)}$$

反射律と推移律を満たす 2 項関係 \leq を前順序 (*preorder*) あるいは擬順序という。反対称律を満たす擬順序は、半順序 (*partial order*) と呼ばれる。さらに、次の規則も考えよう。

$$\text{(比較可能律)} \quad x \leq y \text{ or } y \leq x$$

この比較可能律を満たす半順序は、全順序 (*linear order*) と呼ばれる。擬順序 \leq が整礎 (*well-founded*) であるとは、無限下降列

$$x_1 > x_2 > x_3 > \dots$$

を持たないことを意味する。整礎な全順序は整列順序 (*well-order*) と呼ばれる。そして、順序数 (*ordinal*) とは、整列順序の順序型である。定義から明らかであるように、整列順序 (順序数) とは、あくまである種の《有限性》を持つ順序に過ぎない。

例 2.11. 定義 2.4 の自然数の組上の辞書式順序や、その辞書式順序の入れ子などは整列順序である。

例 2.12. いま、自然数を係数とする 1 変数多項式の比較を考えよう。次数の大きな多項式の方が大きいと考え、また、次数が等しいのであれば、最大次数の項の係数から順に比較していく。より正確には、次のように多項式 p, q が与えられていると仮定しよう。

$$\begin{aligned} p &= a_n x^n + a_{n-1} x^{n-1} + \dots + a_1 x + a_0 \\ q &= b_m x^m + b_{m-1} x^{m-1} + \dots + b_1 x + b_0 \end{aligned}$$

このとき、多項式 p と q と間の順序を以下によって定義する。

$$p < q \iff (n < m) \text{ or } [n = m \text{ and } (\exists i \leq n)(\forall k < i) [a_k = b_k \wedge a_i < b_i]]$$

つまり、 $(a_n, a_{n-1}, \dots, a_1, a_0)$ と $(b_m, b_{m-1}, \dots, b_1, b_0)$ の辞書式順序による比較である。このとき、たとえば、多項式間の順序について、以下が成立している。

$$\begin{aligned} 0 < 1 < 2 < \dots < n < n+1 < \dots < x < x+1 < \dots < x+n < x+n+1 < \dots \\ \dots < 2x < 2x+1 < \dots < 3x < \dots < nx < nx+1 < \dots < (n+1)x < \dots \\ \dots < x^2 < x^2+1 < x^2+6 < 2x^2+1 < 5x^2 < x^3+x^2 < x^4+3x+2 < \dots \end{aligned}$$

例 2.12 の自然数を係数とする 1 変数多項式上の順序では、変数を x と書いていたが、以後は変数を ω で表すことにする。また、自然数係数は変数の左に書くのではなく、右に書くことにしよう。

$$6x^5 + 3x^2 + 4x + 8 \rightsquigarrow \omega^5 \cdot 6 + \omega^2 \cdot 3 + \omega \cdot 4 + 8$$

順序数の無限性と有限性: さて、順序数のもつ《無限性》と《有限性》の二面性が本稿では重要である。つまり、順序数の《無限性》を利用することによって、急増加階層 $f_\omega, f_{\omega+1}, \dots$ を無限に拡張していくことができる。一方で、順序数の《有限性》である整礎性は、計算機科学においては、しばしば、計算アルゴリズムの停止性 (無限ループに陥らないこと) の保証などにしばしば用いられる。つまり、状態遷移を $>$ で表すことにすれば、整礎性によって、

$$x_1 > x_2 > x_3 > \dots > x_n = x_{n+1} = x_{n+2} = \dots$$

となることが保証されるから、状態遷移は有限ステップで停止し、計算結果が確定する。そして、これは有限の値を取る関数を作り出すのに役立つ。急増加階層の理論においても、順序数の《有限

性》を利用し，急増加階層の各レベルにおいて有限的な関数 f_α を作り出す．さらに，順序数記法というものがアルゴリズム的に与えられているとき，この関数 f_α の出力値を計算するアルゴリズムも具体的に与えられるのである．

素朴順序数論の入り口：例 2.12 において，自然数を係数とする 1 変数多項式上の順序について取り扱った．先に述べたように，変数に用いる記号を ω で書き直せば，現在までにたとえば以下のような系列を得られている．

$$\begin{aligned} 0 < 1 < 2 < \dots < n < n+1 < \dots < \omega \\ \omega < \omega+1 < \omega+2 < \dots < \omega+n < \omega+n+1 < \dots < \omega \cdot 2 \\ \omega \cdot 2 < \omega \cdot 3 < \dots < \omega \cdot n < \omega(n+1) < \dots < \omega^2 \\ \omega^2 < \omega^3 < \omega^4 < \dots < \omega^n < \omega^{n+1} < \dots \end{aligned}$$

このアイデアはどんどん拡張していくことができる．たとえば， ω を指数部に載せてもよいだろう．次数 1 の式 ω はどんな自然数（次数 0 の式）よりも大きいのだから， ω^ω は ω^n よりも大きくなると考えるのが妥当である．

$$\begin{aligned} \omega^n < \dots < \omega^\omega < \omega^\omega + 1 < \omega^\omega + \omega < \omega^\omega + \omega^2 < \dots < \omega^\omega + \omega^n < \dots < \omega^\omega \cdot 2 \\ \omega^\omega \cdot 2 < \omega^\omega \cdot 3 < \dots < \omega^\omega \cdot n < \omega^\omega(n+1) < \dots < \omega^\omega \cdot \omega = \omega^{\omega+1} \\ \omega^{\omega+1} < \omega^{\omega+2} < \dots < \omega^{\omega+n} < \omega^{\omega+n+1} < \dots < \omega^{\omega+\omega} = \omega^{\omega \cdot 2} \\ \omega^{\omega \cdot 2} < \omega^{\omega \cdot 3} < \dots < \omega^{\omega \cdot n} < \omega^{\omega(n+1)} < \dots < \omega^{\omega \cdot \omega} = \omega^{\omega^2} \\ \omega^{\omega^2} < \omega^{\omega^3} < \dots < \omega^{\omega^n} < \omega^{\omega^{n+1}} < \dots < \omega^{\omega^\omega} \end{aligned}$$

このアイデアはどんどん拡張していくことが可能である．

$$\omega < \omega^\omega < \omega^{\omega^\omega} < \omega^{\omega^{\omega^\omega}} < \dots < \underbrace{\omega^{\omega^{\dots^\omega}}}_{n \text{ 個}} < \dots$$

我々の道のりはまだこれで終わりではない．この無限の先にはさらに ε_0 と呼ばれる順序数がある．これは， $\omega^\xi = \xi$ を満たす最小の解 ξ として定義される．

$$\underbrace{\omega^{\omega^{\dots^\omega}}}_{n \text{ 個}} < \dots < \omega^{\varepsilon_0} = \varepsilon_0.$$

この概念の正確な定式化については後で説明する．そして，ここからさらに先のステップに進むことができる．

$$\begin{aligned} \varepsilon_0 < \varepsilon_0 + 1 < \varepsilon_0 + \omega^2 < \varepsilon_0 + \omega^\omega < \varepsilon_0 + \omega^{\omega^\omega} < \dots < \varepsilon_0 + \varepsilon_0 = \varepsilon_0 \cdot 2 \\ \varepsilon_0 \cdot 2 < \varepsilon_0 \cdot \omega < \varepsilon_0 \cdot \omega^2 < \varepsilon_0 \cdot \omega^\omega < \varepsilon_0 \cdot \omega^{\omega^\omega} < \dots < \varepsilon_0 \cdot \varepsilon_0 = \varepsilon_0^2 \end{aligned}$$

$$\varepsilon_0^2 < \varepsilon_0^\omega < \varepsilon_0^{\omega^\omega} < \dots < \varepsilon_0^{\varepsilon_0} < \varepsilon_0^{\varepsilon_0^{\varepsilon_0}} < \dots < \underbrace{\varepsilon_0^{\varepsilon_0^{\dots^{\varepsilon_0}}}}_{n \text{ 個}} < \dots$$

この極限として得られるものが， $\omega^\xi = \xi$ を満たす次の解 ε_1 である．

$$\underbrace{\varepsilon_0^{\varepsilon_0^{\dots^{\varepsilon_0}}}}_{n \text{ 個}} < \dots < \omega^{\varepsilon_1} = \varepsilon_1.$$

この ε_1 について注意しておく， ε_1 へと登っていく道として，次のようなルートもある．

$$\varepsilon_0 + 1 < \omega^{\varepsilon_0+1} < \omega^{\omega^{\varepsilon_0+1}} < \omega^{\omega^{\omega^{\varepsilon_0+1}}} < \dots < \underbrace{\omega^{\omega^{\dots^{\omega^{\varepsilon_0+1}}}}}_{n \text{ 個}} < \dots < \varepsilon_1.$$

これはなぜなら，次の式が成立するためである．

$$\omega^{\omega^{\omega^{\varepsilon_0+1}}} = \omega^{\omega^{\omega^{\varepsilon_0} \cdot \omega}} = \omega^{\omega^{\varepsilon_0 \cdot \omega}} = \omega^{(\omega^{\varepsilon_0})^\omega} = \omega^{\varepsilon_0^\omega} = \omega^{\varepsilon_0^{1+\omega}} = \omega^{\varepsilon_0 \cdot \varepsilon_0^\omega} = (\omega^{\varepsilon_0})^{\varepsilon_0^\omega} = \varepsilon_0^{\varepsilon_0^\omega}.$$

さて， ε_1 が得られたら，全く同じ議論を経由して， $\omega^\xi = \xi$ を満たす次の解 ε_2 を得られる．

$$\varepsilon_1 + 1 < \omega^{\varepsilon_1+1} < \omega^{\omega^{\varepsilon_1+1}} < \omega^{\omega^{\omega^{\varepsilon_1+1}}} < \dots < \underbrace{\omega^{\omega^{\dots^{\omega^{\varepsilon_1+1}}}}}_{n \text{ 個}} < \dots < \varepsilon_2.$$

このようにして， $\xi \mapsto \omega^\xi$ の不動点を次々に作っていくことができる．以下， ε_α を $\xi \mapsto \omega^\xi$ の α 番目の不動点を表すことにする．したがって，たとえば ε_α と $\varepsilon_{\alpha+1}$ というたったひとつの隙間にすら，上で述べた途方も無い階層がある．

$$\begin{aligned} \varepsilon_0 < \dots < \omega^{\omega^{\omega^{\varepsilon_0}}} < \dots < \varepsilon_1 < \dots < \omega^{\omega^{\omega^{\varepsilon_1}}} < \dots < \varepsilon_2 < \dots < \omega^{\omega^{\omega^{\varepsilon_2}}} < \dots < \varepsilon_3 < \dots \\ & \dots < \varepsilon_4 < \dots < \varepsilon_n < \dots < \varepsilon_\omega < \dots < \varepsilon_{\omega+1} < \dots < \varepsilon_{\omega \cdot 2} < \dots < \varepsilon_{\omega^2} < \dots \\ & \dots < \varepsilon_{\omega^\omega} < \dots < \varepsilon_{\omega^{\omega^\omega}} < \dots < \varepsilon_{\varepsilon_0} < \dots < \varepsilon_{\varepsilon_0+1} < \dots < \varepsilon_{\varepsilon_0+\omega^\omega} < \dots \\ & \dots < \varepsilon_{\varepsilon_0 \cdot 2} < \dots < \varepsilon_{\varepsilon_0 \cdot \omega^\omega} < \dots < \varepsilon_{\varepsilon_0^2} < \dots < \varepsilon_{\varepsilon_0^{\varepsilon_0}} < \dots < \varepsilon_{\varepsilon_1} < \dots \\ & \dots < \varepsilon_{\varepsilon_2} < \dots < \varepsilon_{\varepsilon_\omega} < \dots < \varepsilon_{\varepsilon_{\omega^\omega}} < \dots < \varepsilon_{\varepsilon_{\varepsilon_0}} < \dots \end{aligned}$$

この操作の適当なステップをピックアップすれば，以下の系列を得られる．

$$\varepsilon_0 < \varepsilon_{\varepsilon_0} < \varepsilon_{\varepsilon_{\varepsilon_0}} < \varepsilon_{\varepsilon_{\varepsilon_{\varepsilon_0}}} < \dots < \underbrace{\varepsilon_{\varepsilon_{\dots_{\varepsilon_0}}}}_{n \text{ 個}} < \dots$$

我々の道のりはまだまだ続く．この極限として得られるものが， $\varepsilon_\xi = \xi$ を満たす最小の解 ζ_0 である．

$$\underbrace{\varepsilon_{\varepsilon_{\dots_{\varepsilon_0}}}}_{n \text{ 個}} < \dots < \varepsilon_{\zeta_0} = \zeta_0.$$

このパターンは延々と続いていく．たとえば， ζ_0 からふたたび類似の議論を繰り返していけば，

$$\zeta_0 < \dots < \omega^{\omega^{\zeta_0}} < \dots < \varepsilon_{\zeta_0+1} < \dots < \omega^{\omega^{\varepsilon_{\zeta_0+1}}} < \dots < \varepsilon_{\zeta_0+2} < \dots < \varepsilon_{\zeta_0+3} < \dots$$

$$\begin{aligned} & \cdots < \varepsilon_{\zeta_0+\omega} < \cdots < \varepsilon_{\zeta_0+\omega^2} < \varepsilon_{\zeta_0+\omega^\omega} < \cdots < \varepsilon_{\zeta_0+\omega^{\omega^\omega}} < \cdots < \varepsilon_{\zeta_0+\varepsilon_0} < \cdots \\ & \cdots < \varepsilon_{\zeta_0+\varepsilon_{\varepsilon_0}} < \cdots < \varepsilon_{\omega^{\zeta_0+1}} < \cdots < \varepsilon_{\omega^{\omega^{\zeta_0+1}}} < \cdots < \varepsilon_{\varepsilon_{\zeta_0+1}} < \cdots < \varepsilon_{\varepsilon_{\varepsilon_{\zeta_0+1}}} < \cdots \end{aligned}$$

この極限として得られるものが、 $\varepsilon_\xi = \xi$ を満たす次の解 ζ_1 である。

$$\zeta_0 + 1 < \varepsilon_{\zeta_0+1} < \varepsilon_{\varepsilon_{\zeta_0+1}} < \varepsilon_{\varepsilon_{\varepsilon_{\zeta_0+1}}} < \cdots < \zeta_1.$$

そして、 ζ_1 が得られたら、全く同じ議論を介して、 $\varepsilon_\xi = \xi$ を満たす次の解 ζ_2 を得られる。

$$\zeta_1 + 1 < \varepsilon_{\zeta_1+1} < \varepsilon_{\varepsilon_{\zeta_1+1}} < \varepsilon_{\varepsilon_{\varepsilon_{\zeta_1+1}}} < \cdots < \zeta_2.$$

このようにして、 $\xi \mapsto \varepsilon_\xi$ の不動点を次々に作っていき、 ζ_α を $\xi \mapsto \varepsilon_\xi$ の α 番目の不動点を表す記号とする。

ヴェブレン階層: さて、そろそろパターンは掴めてきただろうか。 ε_α は $\xi \mapsto \omega^\xi$ の α 番目の不動点であり、 ζ_α は $\xi \mapsto \varepsilon_\xi$ の α 番目の不動点であった。当然、次のレベルとして、 $\xi \mapsto \zeta_\xi$ の不動点を考えることができる。しかし、 $\omega, \varepsilon, \zeta, \dots$ と次のレベルに進むたびに異なるギリシャ文字を選んでいたら、すぐにギリシャ文字が枯渇するので、もう少し一般性のある記法を導入しよう。これからは、 $\omega, \varepsilon, \zeta$ のことは、 ϕ_0, ϕ_1, ϕ_2 と呼び直すことにする。つまり、

$$\phi_0(\alpha) = \omega^\alpha, \quad \phi_1(\alpha) = \varepsilon_\alpha, \quad \phi_2(\alpha) = \zeta_\alpha.$$

そうすると、これらの定義は統一的に書くことができる。

$$\phi_1(\alpha) = \text{“}\phi_0(\xi) = \xi \text{ の } \alpha \text{ 番目の解”}$$

$$\phi_2(\alpha) = \text{“}\phi_1(\xi) = \xi \text{ の } \alpha \text{ 番目の解”}$$

このパターンを延々続けられることは明白だろう。具体的には、たとえば、 ϕ_{n+1} を次のように定義できる。

$$\phi_{n+1}(\alpha) = \text{“}\phi_n(\xi) = \xi \text{ の } \alpha \text{ 番目の解”}$$

このようにして、演算 $\phi_0, \phi_1, \phi_2, \phi_3, \dots, \phi_n, \phi_{n+1}, \dots$ を定義していくことができる。しかし、 ϕ_ω を定義する場合には、注意する必要がある。なぜなら、 ω には直前の数が存在しない。したがって、 ϕ_ω の定義として「直前の演算の不動点を取る」とするのは不適當である。それではどのように ϕ_ω を定義すればよいかというと、その答えは、「これまで作ってきたすべての演算の共通不動点を取る」である。

$$\phi_\omega(\alpha) = \text{“}(\phi_n(\xi) = \xi)_{n < \omega} \text{ の } \alpha \text{ 番目の共通解”}$$

つまり、 $\phi_\omega(\alpha) = \xi$ ならば、以下の等式が成立している。

$$\phi_0(\xi) = \phi_1(\xi) = \phi_2(\xi) = \cdots = \phi_n(\xi) = \phi_{n+1}(\xi) = \cdots = \xi.$$

一般の ϕ_γ の定義としても、「これまで作ってきたすべての演算の共通不動点を取る」という考えを採用できる。つまり、

$$\phi_\gamma(\alpha) = \text{“}(\phi_n(\xi) = \xi)_{\beta < \gamma} \text{ の } \alpha \text{ 番目の共通解”}$$

実際には, $\gamma = \beta + 1$ の場合に関して言えば, 直前の演算の不動点を取ってくれば十分である. ところで, 共通解といっても, どのようにそのような解を得られるかが明確でない. これについては, ε_α や ζ_α を参考にして, 考え直してみることにしよう. まず, $\varepsilon_0 = \phi_1(0)$ への登山ルートとしては, 次のようなものがあった.

$$\begin{aligned} \phi_0(0) = 1 < \phi_0(\phi_0(0)) = \omega < \phi_0(\phi_0(\phi_0(0))) = \omega^\omega < \phi_0(\phi_0(\phi_0(\phi_0(0)))) = \omega^{\omega^\omega} < \dots \\ \dots < \underbrace{\phi_0(\phi_0(\dots(\phi_0(0))))}_{n \text{ 個}} = \underbrace{\omega^{\omega^{\dots^\omega}}}_{n-1 \text{ 個}} < \dots < \phi_1(0) = \varepsilon_0 \end{aligned}$$

$\varepsilon_1 = \phi_1(1)$ へと登っていく道は以下のようなものである.

$$\begin{aligned} \phi_1(0) + 1 = \varepsilon_0 + 1 < \phi_0(\phi_1(0) + 1) = \omega^{\varepsilon_0+1} < \phi_0(\phi_0(\phi_1(0) + 1)) = \omega^{\omega^{\varepsilon_0+1}} \dots \\ \dots < \underbrace{\phi_0(\phi_0(\dots(\phi_0(\phi_1(0) + 1))))}_{n \text{ 個}} = \underbrace{\omega^{\omega^{\dots^{\omega^{\varepsilon_0+1}}}}}_{n-1 \text{ 個}} < \dots < \phi_1(1) = \varepsilon_1 \end{aligned}$$

このパターンをどう続けられればよいかも明らかであろう. たとえば, $\phi_1(n+1) = \varepsilon_{n+1}$ へと登っていく道としては, 次のようなものを考えればよい.

$$\begin{aligned} \phi_1(n) + 1 < \phi_0(\phi_1(n) + 1) < \phi_0 \circ \phi_0(\phi_1(n) + 1) < \dots \\ \dots < \underbrace{\phi_0 \circ \phi_0 \circ \dots \circ \phi_0}_{n \text{ 個}}(\phi_1(n) + 1) < \dots < \phi_1(n+1). \end{aligned}$$

$\phi_1(\omega) = \varepsilon_\omega$ には, 次のようにして登っていくことができる.

$$\phi_1(0) < \phi_1(1) < \phi_1(2) < \dots < \phi_1(n) < \phi_1(n+1) < \dots < \phi_1(\omega)$$

このアイデアは, ϕ の階層の如何なるレベルにも適用できる.

$$0 < \phi_\gamma(0) < \phi_\gamma \circ \phi_\gamma(0) < \dots < \underbrace{\phi_\gamma \circ \dots \circ \phi_\gamma}_{n \text{ 個}}(0) < \phi_{\gamma+1}(0).$$

$$\begin{aligned} \phi_{\gamma+1}(\alpha) + 1 < \phi_\gamma(\phi_{\gamma+1}(\alpha) + 1) < \phi_\gamma \circ \phi_\gamma(\phi_{\gamma+1}(\alpha) + 1) < \dots \\ \dots < \underbrace{\phi_\gamma \circ \phi_\gamma \circ \dots \circ \phi_\gamma}_{n \text{ 個}}(\phi_{\gamma+1}(\alpha) + 1) < \dots < \phi_{\gamma+1}(\alpha + 1). \end{aligned}$$

このようにして得られる演算の階層が得られる.

$$\phi_0, \phi_1, \dots, \phi_n, \phi_{n+1}, \dots, \phi_\omega, \phi_{\omega^\omega}, \phi_{\varepsilon_0}, \phi_{\zeta_0}, \dots, \phi_{\phi_\omega(0)}, \dots$$

この演算の階層 $\{\phi_\gamma\}$ をヴェブレン階層 (*Veblen hierarchy*) と呼ぶ. 演算のレベル γ の部分も, これらの演算を用いて作った順序数で創造していくことにする. しかし, その新たな演算の創造もまた, いつか限界を迎える.

$$\phi_0(0), \phi_{\phi_0(0)}(0), \phi_{\phi_{\phi_0(0)}(0)}(0), \dots$$

この極限を越える順序数をそこまでの階層の演算では生み出すことができない．その極限は， $\phi_\xi(0) = \xi$ となる最小の順序数 ξ である．この順序数は，フェファーマン-シュッテ順序数 (Feferman-Shütte ordinal) と呼ばれ， Γ_0 と書き表される．

$$\phi_{\Gamma_0}(0) = \Gamma_0.$$

多変数ヴェブレン階層： それでは，フェファーマン-シュッテ順序数 Γ_0 が考えうる限界の順序数なのだろうか．その答えは，否である．あくまでフェファーマン-シュッテ順序数 Γ_0 は新たなタイプの不動点にすぎず，不動点があるとき，われわれはさらに先の道へと進むことができる．つまり， Γ_0 とは， $\phi_\xi(0) = \xi$ の最初の解のことであったから，これまでと同様に， $\phi_\xi(0) = \xi$ の解を小さい順に並べていくとしよう．

$$\Gamma_0, \Gamma_1, \Gamma_2, \dots, \Gamma_\omega, \Gamma_{\omega+1}, \dots, \Gamma_{\omega^\omega}, \dots, \Gamma_{\varepsilon_0}, \dots, \Gamma_{\phi(\alpha, 0)}, \dots, \Gamma_{\Gamma_0}, \dots, \Gamma_{\Gamma_1}, \dots, \Gamma_{\Gamma_{\Gamma_0}}, \dots$$

これまで書いてきた ϕ_α のことを $\phi_{0,\alpha}$ という新しい記号で書き直すことにし， $\phi_{1,0}(\alpha) = \Gamma_\alpha$ と定義することにする．つまり，

$$\phi_{1,0}(\alpha) = \text{“}\phi_{0,\xi}(0) = \xi \text{ の } \alpha \text{ 番目の解”}$$

そうすると，ここからふたたびヴェブレン階層と類似の構成をやり直す．最初のヴェブレン階層を“第 0 ヴェブレン階層”と呼ぶことにすれば，次の“第 1 ヴェブレン階層”は，以下のように定義される．

$$\begin{aligned} \phi_{1,1}(\alpha) &= \text{“}\phi_{1,0}(\xi) = \xi \text{ の } \alpha \text{ 番目の解”} \\ \phi_{1,2}(\alpha) &= \text{“}\phi_{1,1}(\xi) = \xi \text{ の } \alpha \text{ 番目の解”} \\ &\vdots \\ \phi_{1,n+1}(\alpha) &= \text{“}\phi_{1,n}(\xi) = \xi \text{ の } \alpha \text{ 番目の解”} \end{aligned}$$

より正確には，これまでのヴェブレン階層の構成のように，ここまで作ってきたすべての関数 $\phi_{i,\gamma}$ の共通不動点を取る，という動作を行う．つまり，

$$\phi_{1,\gamma}(\alpha) = \text{“}(\phi_{1,\delta}(\xi) = \xi)_{\delta < \gamma} \text{ の } \alpha \text{ 番目の共通解”}$$

さて，これは 1 という添字が新たに付加されただけで，元のヴェブレン階層と類似の構成を行っている．したがって，ヴェブレン階層の不動点として Γ_0 を与えたように，この“第 1 ヴェブレン階層”の不動点を取ることができる．

$$\Gamma_0^1 = \text{“}\phi_{1,\xi}(0) = \xi \text{ の最初の解”}$$

また，同様にして， $\phi_{1,\xi}(0) = \xi$ の解を小さい順に並べていくことにすると，次の記法を用いるのが妥当であろう．

$$\phi_{2,0}(\alpha) = \text{“}\phi_{1,\xi}(0) = \xi \text{ の } \alpha \text{ 番目の解”}$$

そして、これを基にして、“第 2 ヴェブレン階層” $\phi_{2,\gamma}$ を作ることができる。この構成は延々、繰り返すことができる。具体的には、第 η ヴェブレン階層は、 $\gamma > 0$ について、以下のように構成される。

$$\phi_{\eta,\gamma}(\alpha) = “(\phi_{\eta,\delta}(\xi) = \xi)_{\delta < \gamma} \text{ の } \alpha \text{ 番目の共通解}”$$

そして、ヴェブレン階層の不動点を取り、新たな階層の始点を作るステップは、次によって得られる。

$$\phi_{\eta,0}(\alpha) = “(\phi_{\theta,\xi}(0) = \xi)_{\theta < \gamma} \text{ の } \alpha \text{ 番目の共通解}”$$

こうして新たな順序数を次々に作り上げることができる。そして、「ヴェブレン階層の階層」の構成でも辿り着けない順序数があり、それが「ヴェブレン階層の階層」の最小不動点である。つまり、次の順序数である。

$$\phi_{1,0,0}(0) = “\phi_{\xi,0}(0) = \xi \text{ の最初の解}”$$

この順序数は、しばしばアッカーマン順序数 (*Ackermann ordinal*) と呼ばれる。さて、 $\phi_{1,0,0}$ という記法のヒントを与えたので、このアッカーマン順序数を更を超える方法は容易に思いつくであろう。もうひとつヒントを与えれば、これまでの $\phi_{\eta,\gamma}$ を $\phi_{0,\eta,\gamma}$ と書き直して考えてみよう。すると、「ヴェブレン階層の階層」の不動点の枚挙は以下によって与えられる。

$$\phi_{1,0,0}(\alpha) = “\phi_{0,\xi,0}(0) = \xi \text{ の } \alpha \text{ 番目の解}”$$

次の段階で構成されるのは、「ヴェブレン階層の階層の階層」である。その構成をまとめて書き表せば、 $\phi_{\lambda,\eta,\gamma}$ は次のようにして定義される。

$$\phi_{\lambda,\eta,\gamma}(\alpha) = \begin{cases} “(\phi_{\lambda,\eta,\beta}(\xi) = \xi)_{\beta < \gamma} \text{ の } \alpha \text{ 番目の共通解}” & \text{if } \gamma > 0 \\ “(\phi_{\lambda,\beta,\xi}(0) = \xi)_{\beta < \eta} \text{ の } \alpha \text{ 番目の共通解}” & \text{else if } \eta > 0 \\ “(\phi_{\beta,\xi,0}(0) = \xi)_{\beta < \lambda} \text{ の } \alpha \text{ 番目の共通解}” & \text{otherwise} \end{cases}$$

これを続けていくと、添字がどんどん複雑になってしまうので、 $\phi_{\lambda,\eta,\gamma}(\alpha)$ の代わりに $\phi(\lambda, \eta, \gamma, \alpha)$ と書くことにする。もちろん、アッカーマン順序数の定義のときと同様にして、 $\phi(\xi, 0, 0, 0) = \phi_{\xi,0,0}(0) = \xi$ の最小解を $\phi(1, 0, 0, 0) = \phi_{1,0,0,0}(0)$ として定義できる。これを繰り返して、新たな順序数を作り上げていく。もちろん、変数の数はどんどん増えていく。

さて、ここまでの構成の法則性は掴めただろうか。上の $\phi_{\lambda,\eta,\gamma}$ の定義を見ると、添字の右から順に見ていき、最初に 0 以外の値が現れた添字に注目して、場合分けをしている。この法則性が掴めたならば、 $\phi(\gamma_n, \gamma_{n-1}, \dots, \gamma_0, \alpha)$ をどのように定義すればよいかの予想も付くはずである。その答えは、 $\gamma_0, \gamma_1, \dots$ と順に見ていき、最初に 0 以外の値が現れた γ_i に注目し、 γ_{i+1} の部分を変数として、不動点を取ればよい。

$$\begin{aligned} &\phi(\gamma_n, \gamma_{n-1}, \dots, \gamma_{i+1}, \gamma_i, 0, 0, \dots, 0, 0, \alpha) \\ &= “(\phi(\gamma_n, \gamma_{n-1}, \dots, \gamma_{i+1}, \beta, \xi, 0, \dots, 0, 0) = \xi)_{\beta < \gamma_i} \text{ の } \alpha \text{ 番目の共通解}” \end{aligned}$$

こうして、有限個の順序数を入力に取る関数 ϕ が定義された。この ϕ は、多変数ヴェブレン関数と呼ばれる。さて、順序数の拡張について、これ以上の拡張は困難のように思える。この限界と

なる順序数，つまり，多変数ヴェブレン関数では辿り着けない最小の順序数は，小ヴェブレン順序数 (*small Veblen ordinal*) と呼ばれ，しばしば $\vartheta(\Omega^\omega)$ と書かれる．

$$\phi(1, 0) < \phi(1, 0, 0) < \phi(1, 0, 0, 0) < \phi(1, 0, 0, 0, 0) < \dots < \vartheta(\Omega^\omega).$$

小ヴェブレン順序数があるならば大ヴェブレン順序数もあるのかと疑問に思うのが自然だろう．実は，上で定義したヴェブレン関数を「順序数個の変数を持つ関数」に拡張することによって，小ヴェブレン順序数をさらに超えていくことができる．こうして作られた超限変数ヴェブレン関数の限界が，大ヴェブレン順序数 $\vartheta(\Omega^\Omega)$ と呼ばれるものである．

本稿ではここまでの順序数しか取り扱わないが，順序数の世界はまだまだ先へと広がっていく．ここまで途方もなく巨大な多変数ヴェブレン階層も，可算順序数のうちの計算可能順序数の中のごく小さな領域を占めるに過ぎない．

$$\Gamma_0 < \vartheta(\Omega^\omega) < \vartheta(\Omega^\Omega) \ll \dots \ll \omega_1^{\text{CK}} \ll \dots \ll \omega_1.$$

ここで， $<$ ではなく \ll や \lll と書いた理由として， $\ll \dots \ll$ の中には，ここまで述べたよりも遙かに広大な世界がその中に眠っていることを示唆するためであり， $\lll \dots \lll$ の中には，それとは比較にならないほど広大な世界が佇んでいる．

2.4. 順序数の基本列と急増加関数の階層

ここからは順序数の形式理論を導入しよう．とりあえず，扱いの簡単な Γ_0 未満の順序数について考察する．つまり， $0, +, \phi$ から構成される順序数のみを扱う．ただし， Γ_0 未満の順序数は， $\phi(\gamma, \alpha) = \phi_\gamma(\alpha)$ の形の関数を用いて作られるので， ϕ は 2 変数に制限されている．

形式的には， Γ_0 未満の順序数の体系 O_V を，ある有限的な算術体系として導入していく．まず，言語として $\{0, +, \phi\}$ を持つ．ここで， 0 は定数記号であり， $+$ と ϕ は 2 変数関数記号である．これらの記号の有限の組み合わせで作られる記号列のみを取り扱う．形式的には，以下の概念を考える．

定義 2.13. 体系 O_V の項 (*term*) は，以下のように帰納的に定義される．

1. 0 は項である．
2. s, t が項ならば， $s + t$ および $\phi(s, t)$ も項である．

ちなみに，バックス・ナウア記法 (BNF) と呼ばれるものを用いると，上の項の定義は，以下のように書き直せる．

$$t ::= 0 \mid t + t \mid \phi(t, t)$$

つまり， O_V -項は， $0, +, \phi$ を使って作られる有限長の記号列に過ぎない．さて， O_V -項がいかなる性質を持つかを議論しなければならない．以下が O_V -項の満たす公理である．

1. $+$ は 0 を単位元とする O_V -項のモノイド演算である:

$$(s + t) + u = s + (t + u) \quad t + 0 = 0 + t = t$$

2. 0 は \leq に関する O_V -項の最小元であり, また, $+$ は順序を保存する:

$$0 \leq t \quad \frac{s \leq t \quad u \leq v}{s + u \leq t + v}$$

3. 関数記号 ϕ は順序を保存し, 前の演算の不動点を与える.

$$t \leq \phi(s, t) \quad \frac{s < \phi(t, u)}{s + \phi(t, u) = \phi(t, u)}$$

$$\frac{u \leq v \quad s \leq t}{\phi(u, s) \leq \phi(v, t)} \quad \frac{s < t}{\phi(s, \phi(t, u)) = \phi(t, u)}$$

以上, Γ_0 未満の順序数の体系 O_V は, このような有限的な項と有限的な公理からなる, 有限的な体系である. 次の形の項は, カントール標準形 (Cantor normal form) と呼ばれる.

$$\phi(s_0, t_0) + \phi(s_1, t_1) + \cdots + \phi(s_n, t_n).$$

ただし, ここで,

$$\phi(s_0, t_0) \geq \phi(s_1, t_1) \geq \cdots \geq \phi(s_n, t_n), \quad \phi(s_i, t_i) > t_i$$

であり, s_i, t_i もまた標準形で与えられているものとする. カントール標準形の最終項 $\phi(s_n, t_n)$ が $\phi(0, 0)$ であるとき, この項は後続 (successor) であるといい, さもなくば極限 (limit) であるという.

標準形の中の等しい項は, 掛け算を使ってまとめることができる. ただし, O_V -項は, 掛け算記号を含まないが, $+$ の繰り返しの略記として, 掛け算記号を導入する.

$$t \cdot n := \underbrace{t + t + \cdots + t}_{n \text{ 個}}$$

一応, 注意しておく, 項 s, t に関する積 $s \cdot t$ などは現時点では考えていない. あくまで項 t と自然数 n について, $t \cdot n$ が定義されるだけである. 以後, $\phi(0, \alpha)$ を ω^α と略記し, 特に $\omega^0 = 1$ と略記する. また, $1 \cdot n$ は n と略記される.

この掛け算記号を用いると, 項のカントール標準形は, さらに次の形に書き直せる.

$$\phi(s_0, t_0) \cdot k_0 + \phi(s_1, t_1) \cdot k_1 + \cdots + \phi(s_n, t_n) \cdot k_n$$

$$(\phi(s_0, t_0) > \phi(s_1, t_1) > \cdots > \phi(s_n, t_n), \quad \phi(s_i, t_i) > t_i).$$

ここで, s_i, t_i もまた標準形で与えられている. 以後, 項は常にカントール標準形で与えられていると仮定する.

順序数の梯子: われわれは順序数の山に登りたいのだから, 各項 (順序数) に対して, そこまで登っていくためのルートを開拓する必要がある. われわれは始点 0 のみを知っており, 他の項 t へと登るルートはすべて自身で作り出す必要がある. 項 t へ登るルート $(a_n)_{n \in \mathbb{N}}$ として求められる条件は, 以下である.

$$\begin{aligned} m \leq n &\implies a_m \leq a_n < t, \\ s < t &\implies (\exists n) s \leq a_n. \end{aligned}$$

つまり, $(a_n)_{n \in \mathbb{N}}$ は t の下で非有界な単調増大列である. このような条件を満たす項の列 $(a_n)_{n \in \mathbb{N}}$ は, t の梯子 (ladder) または基本列 (fundamental sequence) と呼ばれ, 梯子 (基本列) の第 n 段目 a_n のことをしばしば $t[n]$ と書く. 第 2.3 節に倣って, 項 t への基本列を帰納的に作ってみよう.

(a) 最初の梯子: まずは, 項 $\phi(0, 0) = 1$ に登るための梯子は自明であり, 以下によって与えられる.

$$0 \leq 0 \leq 0 \leq \dots \leq 0 \leq 0 \leq \dots < 1.$$

つまり, $\phi(0, 0)[n] = 0$ であり, これが梯子の条件を満たすことは容易に確認できる.

(b) 多項式へ登るための梯子: 項は常にコントロール標準形で与えられているという仮定より, 項 t は常に次のような形である.

$$t = \phi(u_0, v_0) + \phi(u_1, v_1) + \dots + \phi(u_\ell, v_\ell).$$

後続順序数の場合には $\phi(u_\ell, v_\ell) = 1$ であることに注意しよう. まず, t が複数の部分項の和である多項式である, つまり $\ell > 0$ である場合を考えよう. 帰納的に, 項の部分項には既に梯子が掛けられていると仮定する. 特に, $\phi(u_\ell, v_\ell)$ には既に梯子 $(\phi(u_\ell, v_\ell)[n])_{n \in \mathbb{N}}$ が掛けられている. このとき, t へは次のように梯子を掛けることができる.

$$t[n] = \phi(u_0, v_0) + \phi(u_1, v_1) + \dots + \phi(u_\ell, v_\ell)[n].$$

後続順序数の場合には $\phi(u_\ell, v_\ell) = 1$ であったから, (a) より $\phi(u_\ell, v_\ell)[n] = 0$ となる. つまり, 常に $(\alpha + 1)[n] = \alpha$ になっている.

後は, t が単項式, つまり $t = \phi(u, v)$ の形である場合を考えればよい.

(c) 指数を 1 段登るための梯子: つづいて, 項 $\phi(0, t + 1) = \omega^{t+1}$ に登るための梯子は, 以下によって与えられる.

$$1 < \omega^t < \omega^t \cdot 2 < \omega^t \cdot 3 < \dots < \omega^t \cdot n < \omega^t \cdot (n + 1) < \dots < \omega^{t+1}.$$

つまり, $\phi(0, t + 1)[n] = \phi(0, t) \cdot n$ によって定義すると, これは梯子の条件を満たしている.

(d) 階層の次のレベルへ登るための梯子: 項 $\phi(s + 1, 0)$ へ登るための梯子について考えよう. この梯子は, 前節のヴェブレン階層の項目で既に作っている.

$$0 < \phi_\gamma(0) < \phi_\gamma \circ \phi_\gamma(0) < \dots < \underbrace{\phi_\gamma \circ \dots \circ \phi_\gamma(0)}_{n \text{ 個}} < \phi_{\gamma+1}(0).$$

形式的には， $\phi(s+1, 0)$ へ登るための梯子は以下によって定義する．

$$\phi(s+1, 0)[0] = 0, \quad \phi(s+1, 0)[n+1] = \phi(s, \phi(s+1, 0)[n]).$$

(e) 次の不動点への梯子：項 $\phi(s+1, t+1)$ へ登るための梯子もまた，前節のヴェブレン階層の項目で既に作っている．

$$\begin{aligned} \phi_{\gamma+1}(\alpha) + 1 &< \phi_{\gamma}(\phi_{\gamma+1}(\alpha) + 1) < \phi_{\gamma} \circ \phi_{\gamma}(\phi_{\gamma+1}(\alpha) + 1) < \dots \\ &\dots < \underbrace{\phi_{\gamma} \circ \phi_{\gamma} \circ \dots \circ \phi_{\gamma}}_{n \text{ 個}}(\phi_{\gamma+1}(\alpha) + 1) < \dots < \phi_{\gamma+1}(\alpha + 1). \end{aligned}$$

形式的には， $\phi(s+1, t+1)$ へ登るための梯子は以下によって定義する．

$$\phi(s+1, t+1)[0] = \phi(s+1, t) + 1, \quad \phi(s+1, t+1)[n+1] = \phi(s, \phi(s+1, t+1)[n]).$$

(f) 極限番目の不動点への梯子：極限項 t について，項 $\phi(s, t)$ へ登るための梯子を与える．帰納的に，この部分項 t へ登る梯子 $(t[n])_{n \in \mathbb{N}}$ は既に掛けられていると仮定できる．このとき，項 $\phi(s, t)$ へは次のように登っていくことができる．

$$\phi(s, t[0]) < \phi(s, t[1]) < \phi(s, t[2]) < \dots < \phi(s, t[n]) < \phi(s, t[n+1]) < \dots < \phi(s, t).$$

つまり， $\phi(s, t)[n] = \phi(s, t[n])$ と定義する．

(g) 階層の極限レベルへの梯子：極限項 s について，項 $\phi(s, 0)$ へ登るための梯子を与える．帰納的に，この部分項 s へ登る梯子 $(s[n])_{n \in \mathbb{N}}$ は既に掛けられていると仮定できる．このとき，項 $\phi(s, 0)$ へは次のように登っていくことができる．

$$\phi(s[0], 0) < \phi(s[1], 0) < \phi(s[2], 0) < \dots < \phi(s, 0).$$

つまり， $\phi(s, 0)[n] = \phi(s[n], 0)$ と定義する．

(h) 極限レベルにおける次の不動点への梯子：極限項 s について，項 $\phi(s, t+1)$ へ登るための梯子を与える．帰納的に，この部分項 s へ登る梯子 $(s[n])_{n \in \mathbb{N}}$ は既に掛けられていると仮定できる．このとき，(e) と同様にして，項 $\phi(s, t+1)$ へは次のように登っていくことができる．

$$\phi(s, t+1)[0] = \phi(s, t) + 1, \quad \phi(s, t+1)[n+1] = \phi(s[n], \phi(s, t+1)[n]).$$

定義 2.14. O_V の梯子系 (*ladder system*) とは，任意の O_V -項 t に対する梯子 $(t[n])_{n \in \mathbb{N}}$ の割り当てを意味する．

個々の項には無数の梯子があり得るが，そのうちのひとつを指定しておくものが梯子系である．特に，梯子系は一意ではないことに注意する．上記の (a)–(h) の操作によって与えられる梯子系を標準的な梯子系と呼ぶことにする．

問題 8. 以下の 2 つの値を求めよ .

$$\phi(\omega, 1)[3], \quad (\phi(\omega^4 + 1, 0)[1])[3].$$

急増加関数の階層: 第 2.1 節において, 急増加関数の階層を定義していた .

$$\mathbf{f}_0, \mathbf{f}_1, \mathbf{f}_2, \mathbf{f}_3, \dots, \mathbf{f}_n, \mathbf{f}_{n+1}, \dots, \mathbf{f}_\omega, \mathbf{f}_{\omega+1}, \dots, \mathbf{f}_{\omega+n}, \mathbf{f}_{\omega+n+1}, \dots$$

梯系を用いることによって, この急増加関数の階層をさらに拡張していくことができる . 具体的には, 項の梯系が指定されているとき, 梯子を辿ることによって, 以下のように急増大関数の階層を定義することができる .

定義 2.15. 与えられた O_V の梯系の下で, 各 O_V -項 t に対して, 関数 $\mathbf{f}_t: \mathbb{N} \rightarrow \mathbb{N}$ を以下のように帰納的に定義する .

$$\begin{aligned} \mathbf{f}_0(n) &= n + 1, \\ \mathbf{f}_{t+1}(n) &= \underbrace{\mathbf{f}_t \circ \mathbf{f}_t \circ \dots \circ \mathbf{f}_t}_{n \text{ 個}}(n), \\ \mathbf{f}_s(n) &= \mathbf{f}_{s[n]}(n) \quad (s \text{ が極限の場合}). \end{aligned}$$

ところで, $\mathbf{f}_{\omega+1}$ 程度でグラハム数を定義するほどの急増加関数となるのであった . そして, 添字の順序数の値を $+1$ していくだけで, 以前までの関数とは比べ物にならないほどの急増加関数となる . したがって, たとえば

$$\mathbf{f}_{\omega \cdot 2}, \mathbf{f}_{\omega^\omega}, \mathbf{f}_{\varepsilon_0}, \mathbf{f}_{\varepsilon_1}, \mathbf{f}_{\zeta_0}, \mathbf{f}_{\zeta_\omega}, \mathbf{f}_{\phi(3,0)}, \mathbf{f}_{\phi(\omega,0)}, \mathbf{f}_{\phi(\varepsilon_0,0)}, \dots$$

などの関数の増大速度は想像を絶するものであろう . 本稿では省略するが, 梯系は多変数ヴェブレン階層 (あるいはその先) にも拡張することもでき, 以下のような急増加関数を定義することも可能である .

$$\mathbf{f}_{\Gamma_0}, \mathbf{f}_{\phi(1,0,0,0)}, \mathbf{f}_{\vartheta(\Omega^\omega)}, \mathbf{f}_{\vartheta(\Omega^\Omega)}$$

2.5. 木の埋め込み順序によるヴェブレン階層の表示

多変数ヴェブレン階層は途方もなく巨大な階層のように思える . このような尋常でなく巨大なものが, 通常の数学の文脈で到底出てくとは思えない . しかし, 驚くべきことに, 「有限木の \wedge -埋め込み」という極めて単純で有限的な構造の中に, この多変数ヴェブレン階層という超巨大な階層が埋まっていることが発見されたのである !

O_V -項で表されるのは Γ_0 未満の順序数のみであるが, 以後は, O_V -項のことを単に順序数 (*ordinal*) と呼んでしまうことにする .

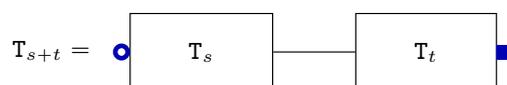
有限木による順序数の表示: さて, ここからは順序数 t に対して木 \mathbb{T}_t を割り当てることにしよう . 第 2.2 節では, 列の対 $(\bar{u}; \bar{v})$ によって, $\omega^{\omega \cdot 2}$ 未満の順序数を表し, それに対応する木 $[\bar{u}; \bar{v}]$ を

定義した．そのアイデアを拡張した木 T_t の割り当て方もあるのだが，それには順序数の積の知識が必要である．本稿では，順序数の取り扱いを最小限に留めるため，積を用いないような，木の割り当てを導入する．以下，あらゆる順序数は，カントール標準形で与えられていると仮定する．

さて，本稿では便宜上，木は空集合ではないとしてきたが，順序数 0 には空集合 \emptyset を対応させるのが妥当である．したがって，ここからは空集合も木として認めることにしよう．

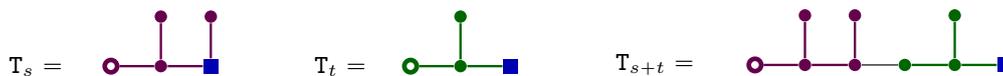
$$T_0 = \emptyset$$

(a) 足し算 $+$ の木表示: 足し算 $+$ は，第 2.2 節の考え方と同様である．つまり，単に直列につなげていくだけである．具体的には，項 $s+t$ に対する木 T_{s+t} は以下のように構成される．

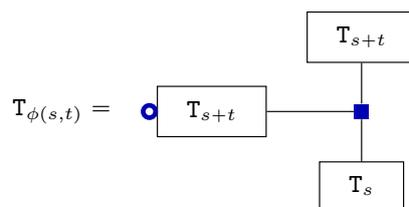


しかし，木を直列に繋げるときに， T_s の根と T_t のノードを接続する必要があるが，接続候補のノードは複数ありえる．そこで，木には常に接続用のノードを 1 つ指定しておくことにする．それが上図の穴あき丸で表した部分である．

もう少し，具体的な例を挙げてみれば，以下のような感じである．



(b) ヴェブレン関数 ϕ の木表示: つづいて， $\phi(s, t)$ に対応する木 $T_{\phi(s, t)}$ を定義しよう．これは，次のようにして定義される．

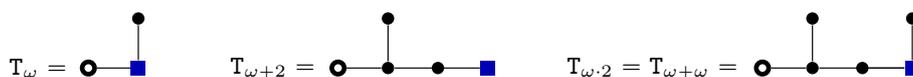


ここで，青四角でハイライトしている根は，木 T_s, T_{s+t}, T_{s+t} の根と接続されている．

▶ ω 未満の順序数: 具体例から順に説明していこう． $\phi(0, 0) = \omega^0 = 1$ であり， T_0 は空集合であったことを思い出す．したがって， T_1 は一点からなる木であり， T_n は頂点数 n の直線状の木である．

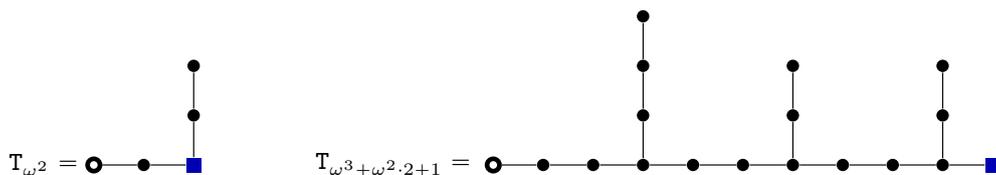
$$T_1 = \blacksquare \quad T_2 = T_{1+1} = \bullet - \blacksquare \quad T_3 = T_{2+1} = \bullet - \bullet - \blacksquare$$

▶ ω^2 未満の順序数: つづいて， $\phi(0, 1) = \omega$ に対応する木は，根に反時計周りに T_1, T_1, T_0 を接続すればよいのだから，以下のような形状をしている．



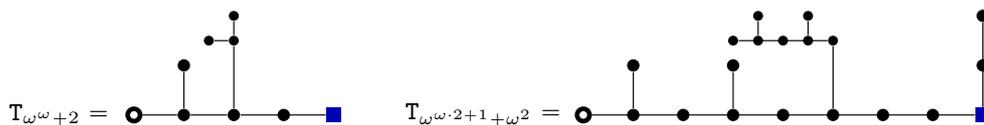
これは、第 2.2 節で導入した方法とは異なることに注意する。順序数を木によって表記する方法はいくらでもあり、いかなる表記法も絶対的なものではない。このレベルの木は分岐を持つので、前のレベルの木には \wedge -埋め込み不可能である。このレベルの木の埋め込み不可能性と順序数の順序の対応は、補題 2.5 と同様にして証明できる。

▶ ω^ω 未満の順序数: $\phi(0, 2) = \omega^2$ の場合には、根に反時計周りに T_2, T_2, T_0 を接続していく。 $\phi(0, 3) = \omega^3$ であれば、 T_3, T_3, T_0 である。



ふたたび、第 2.2 節で導入した方法とは異なることに注意しておく。第 2.2 節のように、積を利用した木の構成の方が頂点数が少なく経済的である。指数部が大きくなるたびに、分岐の先の頂点数が増えていく。足し算 $+$ による結合では、分岐のうちのひとつしか、その先の頂点数を増やさないことを注意しよう。たとえば、 T_{ω^n} を別の木 T に埋め込むためには、 T の 2 分岐の両方の先に必ず長さ n 以上の頂点を持たなければならない。したがって、高い指数の木を低い指数の木に埋め込むことはできない。

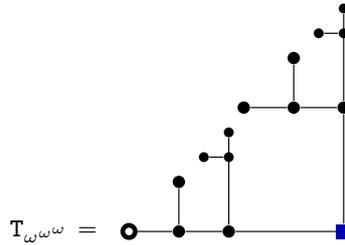
▶ ω^{ω^ω} 未満の順序数: 次の例として、項 $\phi(0, \phi(0, 1))$ つまり $\phi(0, \omega) = \omega^\omega$ を考えると、根に反時計周りに T_ω, T_ω, T_0 を接続し、項 $\phi(0, \phi(0, 1) + \phi(0, 1) + 1)$ つまり $\phi(0, \omega \cdot 2 + 1) = \omega^{\omega \cdot 2 + 1}$ を考えると、根に反時計周りに $T_{\omega \cdot 2 + 1}, T_{\omega \cdot 2 + 1}, T_0$ を接続する。



このレベルの木では、分岐の先でもう一度分岐する、ということが起きる。このために、このレベルの木を以前のレベルの木に埋め込むことはできない。しかし、このレベルでは、分岐の先の分岐のさらに先の分岐は起こらない。第 2.2 節と木の形状は異なるが、このレベルまでの木を利用して、グラハム数やアッカーマン関数などを越えることができる。

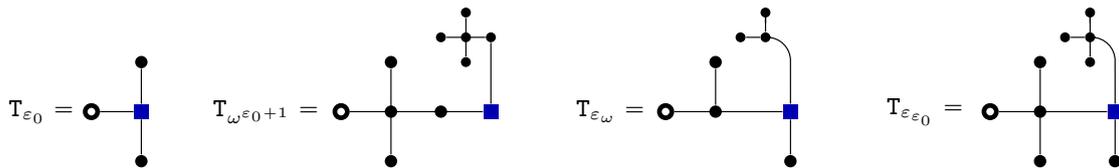
▶ ε_0 未満の順序数: さらに先に進むと、分岐の先で分岐が起こり、さらにまた分岐が起こる、といったように、だんだんと木らしくなってくる。たとえば、項 $\phi(0, \phi(0, \phi(0, 1)))$ つまり $\phi(0, \omega^\omega) = \omega^{\omega^\omega}$

であれば，根に反時計周りに $T_{\omega\omega}, T_{\omega\omega}, T_0$ を接続する．



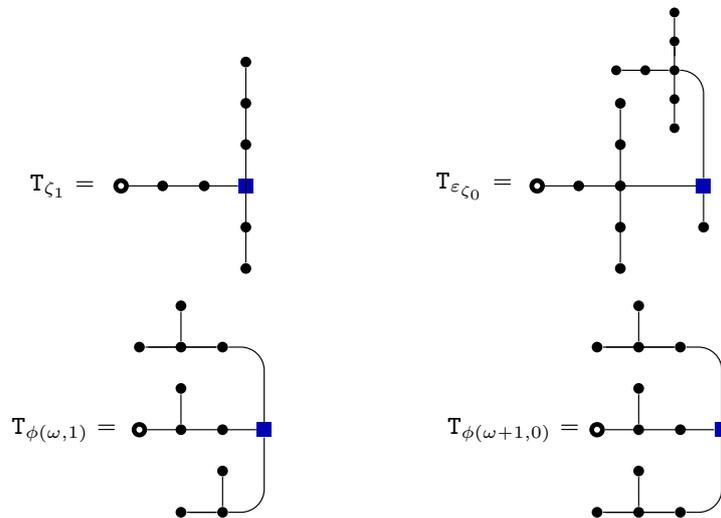
一般に， $\omega \uparrow n$ に対応する木は高さ $n+1$ の完全 2 分木になることがわかる．この方法を用いると，分岐数 2 以下のノードしか持たない木によって， ε_0 未満の順序数はすべて表されるのである．

▶ ζ_0 未満の順序数： それでは， ε_0 の壁を越えてみよう．最初に考えるものは， $\phi(1, 0) = \varepsilon_0$ であるが，これは根に反時計周りに T_1, T_1, T_1 を接続すればよい．他にも， $\phi(0, \phi(1, 0) + 1) = \omega^{\varepsilon_0+1}$ や $\phi(1, \phi(0, 1)) = \varepsilon_\omega$ ，そして $\phi(1, \phi(1, 0)) = \varepsilon_{\varepsilon_0}$ を考えてみよう．



このように，ついに 3 分岐ノードが現れる．このため，2 分岐までしか現れなかった以前のレベルの木には埋め込むことができない．ただし，3 分岐のいずれか 1 つは，その先に頂点数が 1 しかないという点に注目しておこう．

▶ それ以上の順序数： 後は全く同じパターンが続くのみである． $\phi(2, 1) = \zeta_1$ ， $\phi(1, \phi(2, 0)) = \varepsilon_{\zeta_0}$ ， $\phi(\phi(0, 0), 1) = \phi(\omega, 1)$ ， $\phi(\phi(0, 0) + 1, 0) = \phi(\omega + 1, 0)$ 辺りだけ紹介しておくことにしよう．たとえば， ζ_1 であれば，反時計回りに T_3, T_3, T_2 を根に接続し， $\phi(\omega, 1)$ であれば，反時計周りに $T_{\omega+1}, T_{\omega+1}, T_\omega$ を根に接続する．

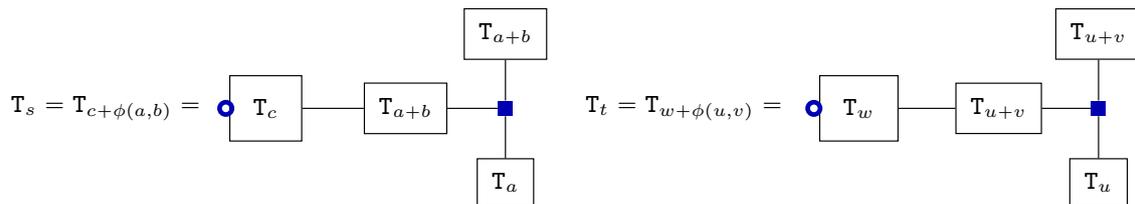


この構成において、 ζ_0 以上の順序数では、先の頂点数がいずれも 2 以上であるような 3 分岐を持つから、これまでのレベルの木に埋め込むことはできない。 $\phi(\omega, 0)$ 以上の順序数では、ある 3 分岐ノードのすべての分岐先でさらなる分岐を持つから、これまでのレベルの木に埋め込むことはできない。といったように、低い順序数の木には埋め込めないような構成を行っている。こうして、高々 3 分岐しか持たない木によって Γ_0 未満の順序数を表示することができる。実際に、その証明を与えてみよう。

定理 2.16. s, t を O_V -項 (つまり Γ_0 未満の順序数) とする。このとき、以下が成立する。

$$T_s \leq T_t \implies s \leq t.$$

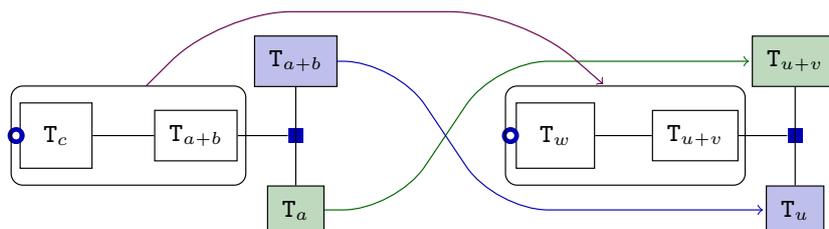
証明. 項の構成に関する帰納法による。項 s または t のどちらかが 0 のときは明らかである。つづいて、 $s = c + \phi(a, b)$ であり、 $t = w + \phi(u, v)$ の形であると仮定する。木 T_s を見ると、0 時の方角から反時計周りに T_{a+b}, T_{c+a+b}, T_a が根に接続されている。木 T_t については、反時計周りに T_{u+v}, T_{w+u+v}, T_u が根に接続されている。



\wedge -埋め込み $h: T_s \rightarrow T_t$ が与えられているとする。木 T_s の根の h による行き先が T_t の根でない場合には、 T_s は、 T_{u+v}, T_{w+u+v}, T_u のいずれかに埋め込まれている。つまり、 T_s はより単純な木 (t の部分項から得られる木) に埋め込まれているので、帰納的仮定に還元される。

したがって、埋め込み h は T_s の根を T_t の根に移すと仮定できる。ここからは根の分岐数が 3 という前提の上で解説するが、2 以下でも同様の議論である。根を根に移している場合には、下限 \wedge を保つという性質より、 T_s の根の分岐先 T_{a+b}, T_{c+a+b}, T_a は、それぞれ T_t の根の異なる分岐先 T_{u+v}, T_{w+u+v}, T_u に埋め込まれている必要がある。

まず、埋め込みの上下反転が行われている場合を考えよう。つまり、 T_{a+b} が T_u に埋め込まれて、 T_a が T_{u+v} に埋め込まれる場合である。

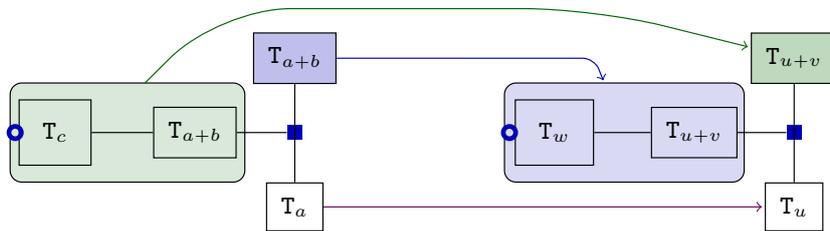


しかし、この仮定を和 + に対する木の定義と組み合わせると、明らかに

$$T_a \leq T_{a+b} \leq T_u, \quad T_{a+b} \leq T_u \leq T_{u+v}$$

であるから、 T_a を T_u に埋め込み、 T_{a+b} を T_{u+v} に埋め込み直すことができる。つまり、上下反転を戻しても、 \wedge -埋め込みになっている。したがって、上下反転はすべて正常位置に戻して、 \wedge -埋め込みは上下反転を含まないと仮定できる。

つづいて、埋め込みの上下反転以外の位置交換が起きている場合について考察する。たとえば、以下のように、根の左側と上側の交換が起きているパターンがあり得る。



上はあくまで例であるが、上下反転以外の位置交換が起きている場合には、必ず次の埋め込みが発生していることを確認できる。

$$T_a \leq T_u, \quad T_{a+b} \leq T_{u+v}, \quad T_c \leq T_{u+v} \quad (2)$$

たとえば、上の例においては、左部から上部への埋め込み $T_{c+a+b} \leq T_{u+v}$ から 2 番目と 3 番目の不等式を導くことができ、下部の埋め込みが 1 番目の不等式に対応する。最初の 2 つの不等式に帰納的仮定を適用すれば、 $a \leq u$ かつ $a+b \leq u+v$ が導かれる。もし $a = u$ ならば、 $b \leq v$ を得られるので

$$\phi(a, b) = \phi(u, b) \leq \phi(u, v)$$

である。もし $a < u$ かつ $b < \phi(u, 0)$ であれば、

$$\phi(a, b) < \phi(a, \phi(u, 0)) = \phi(u, 0) \leq \phi(u, v)$$

を得る。いま、 $a < u$ かつ $b \geq \phi(u, 0)$ であれば、 $u < \Gamma_0$ であるから、 $a \leq u < \phi(u, 0) \leq b$ なので、 $u + \phi(u, 0) = \phi(u, 0)$ より、 $a+b = b$ かつ $u+v = v$ を得るから $b \leq v$ である。したがって、 $a < u$ かつ $b \leq v$ であるから、

$$\phi(a, b) \leq \phi(u, v).$$

いずれにせよ $\phi(a, b) \leq \phi(u, v)$ を得る。つづいて、式 (2) 中の 3 番目の不等式に帰納的仮定を適用すれば、 $c \leq u+v \leq \phi(u, v)$ を得る。よって、 ϕ の性質より、

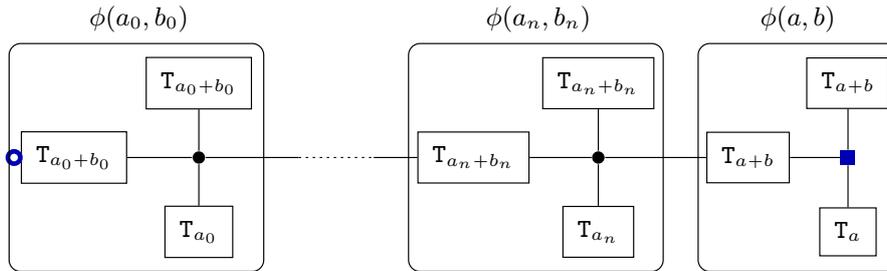
$$c + \phi(a, b) \leq \phi(u, v) \leq w + \phi(u, v)$$

である。よって、埋め込みが上下反転以外の位置交換を伴う場合には定理は示される。また、埋め込みが回転や反転を行わない場合でも、 $T_{c+\phi(a,b)} \leq T_{\phi(u,v)}$ の場合には不等式 (2) が成立している。

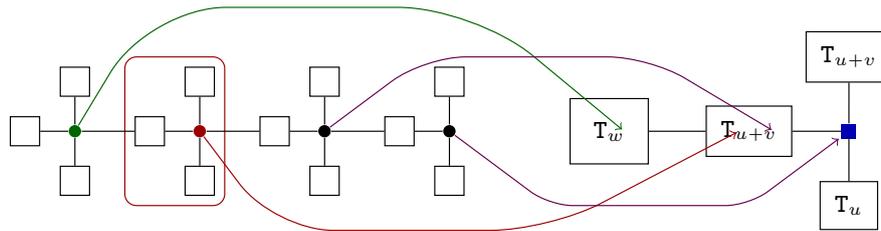
それでは、埋め込み h が回転や反転を含まず、 $T_{c+\phi(a,b)} \preceq T_{\phi(u,v)}$ の場合を考えよう。いま、カントール標準形を考えているので、実際には、 ϕ 型の木が直列に結合した形を考えている。

$$s = c + \phi(a, b) = \phi(a_0, b_0) + \phi(a_1, b_1) + \cdots + \phi(a_n, b_n) + \phi(a, b)$$

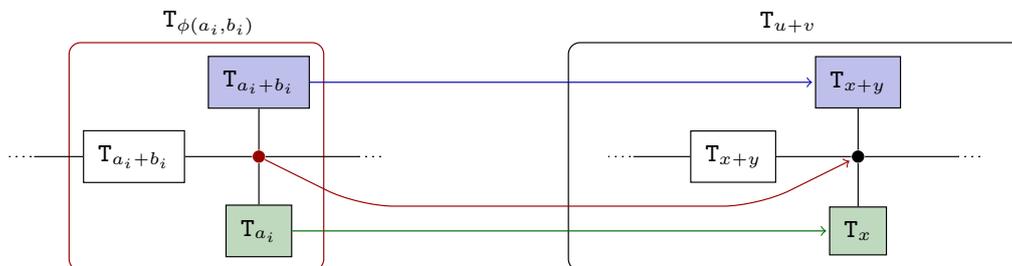
図で表せば、以下のような形状である。



いま、各 ϕ 型の木の根の部分に注目し、その根の h による行き先が $T_{\phi(u,v)}$ に含まれるような最も左のものに注目する。

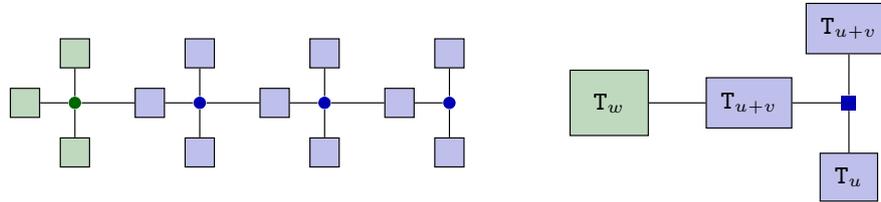


この赤色でハイライトした ϕ 型の木が $T_{\phi(a_i, b_i)}$ であると仮定しよう。また、この部分までの ϕ 型の木は、すべて正常な向きで埋め込まれていると仮定してよい。正常な向きでなければ、先ほどと同様に不等式 (2) が成立する。この ϕ 型の木が一番右側の木でなければ、赤色ノードの埋め込み h の行き先は、 T_{u+v} の中の 3 分岐ノードである。この 3 分岐ノードの上部の木を T_{x+y} とし、下部の木を T_x とする。左部には長く木が連なっている可能性もあるが、順序数に対応する木の構成においては、最初に T_{x+y} が現れてから、その後の木が続くはずである。したがって、埋め込みが正常な向きであることから、以下のような状況になっている。



いま、分岐の上部に注目すると、 $T_{a_i+b_i} \leq T_{x+y}$ であるから、分岐の左部の $T_{a_i+b_i}$ も T_{x+y} に埋め込むことができる。したがって、元の埋め込みをそのような埋め込みに修正することによって、

赤枠部 $T_{\phi(a_i, b_i)}$ 全体が T_{u+v} に埋め込まれていると仮定できる．このようにして，埋め込み先が $T_{\phi(u, v)}$ を外れる最初のノードが， ϕ 型の木の直列の区切り部分となることを保証できる．



以上をまとめると， $s = s_0 + s_1$ となる項 s_0, s_1 が存在して， $T_{s_0} \leq T_w$ かつ $T_{s_1} \leq T_{\phi(u, v)}$ となる．つまり，上図で説明すれば， s_0 が左の木の緑色の部分であり， s_1 が左の木の青色の部分である．よって，帰納的仮定より， $s_0 \leq w$ かつ $s_1 \leq \phi(u, v)$ となるから，和の性質より

$$s = s_0 + s_1 \leq w + \phi(u, v) = t$$

を得る．上の議論では， $T_{\phi(a_i, b_i)}$ が一番右の ϕ 型の木でないことを仮定していたが，一番右であっても証明は同様である．□

ところで，tree 関数の良い下界を求めたい場合には，少ない頂点数で大きい順序数を表示できたほうがよい．第 2.2 節の方法を拡張したタイプの順序数の木表示法をゲーゴロジスト [1] は提示しており，これは本稿で採用したものよりも頂点数の節約という点で優れている．しかし，その代償として，ゲーゴロジストの木表示法を用いると，定理 2.16 の証明がかなり面倒になると思われる（というか 2020 年 4 月現在，書き下された証明は存在しないと思う）．逆に，80 年代～90 年代に書かれた証明論の文献 [15, 5] では，順序数の木表示としていずれとも異なるものを用いており，そのおかげで，本稿と比べると定理 2.16 の証明が圧倒的に容易である．その一方で，ヴェブレン階層程度で 4 分岐が必要になってしまうなど，若干，効率が悪い．証明の容易さを取るか，木の複雑性コスト削減を取るか．この辺りで，証明こそが本質と考える数学者と，一切の証明を行わないゲーゴロジストの対比が出ているようで興味深い．本稿は，数学的証明も巨大数も取り扱いたいため，その中間的な木表示法を採用した．

さて，3 分岐ノードを用いて，ヴェブレン階層の木表示を与えることができたが，この分岐数を増やすことによって，多変数ヴェブレン階層の木表示を与えることも可能である．これを利用すると， $\vartheta(\Omega^\omega)$ の適当な梯系について，tree の増大度が $f_{\vartheta(\Omega^\omega)}$ 程度であると予想できる．

$$\text{tree} \approx f_{\vartheta(\Omega^\omega)}$$

ゲーゴロジストによる記述を見ると，tree の増大度は， $\vartheta(\Omega^\omega)$ の「標準的な梯系」に対する急増加関数 $f_{\vartheta(\Omega^\omega)}$ 程度であることを示したと主張しているようであるが，その証明をどこにも見つけることはできなかった．その後書かれた別のゲーゴロジストによる記述だと，「標準的か分からない特定の梯系に関して，急増加関数 $f_{\vartheta(\Omega^\omega)}$ 程度である」と読める文章もあるので，どちらがゲーゴロジストの現在の認識かは分からない．しかし，どちらにせよ，主張しているだけで証明は一切

書き下されていない(定理 2.16 の証明すら、本稿以前にどこかに書かれているのか定かでない) ようである。

とにかく $\text{tree} \approx \mathbf{f}_{\theta(\Omega^\omega)}$ を証明するには多大な労力を要するであろうが、以下の不等式程度であれば、気合いを入れて頑張れば証明できるので、チャレンジしてみしてほしい。

問題 9. 十分大きな自然数 n について、 $\text{tree}(n) > \mathbf{f}_{\omega^\omega}(n)$ であることを証明せよ。

§ 3. TREE(3) はなぜ有限なのか？

さて、実際に TREE(3) や大きな n について $\text{tree}(n)$ を求めようと試みると、いくらでも長く \wedge -埋め込み不可能列を続けられそうな錯覚に陥る。実際、その巨大さを考えれば、自分の人生が終わるまで \wedge -埋め込み不可能列を続けることも容易いであろう。そうすると、本当に $\text{tree}(n)$ や TREE(n) は有限なのか、 \wedge -埋め込み不可能列は本当に必ず有限ステップで止まるのか。慎重な読者ならば、疑わしいと思うかもしれない。それが数学においては正しい態度である。数学を志すものならば、証明が与えられていない主張を無批判に受け入れてはいけない。証明が書かれていない主張はすべて疑ってかかり、その嘘を暴こうとする心構えを持つことが重要である。

それでは本節では、 $\text{tree}(n)$ や TREE(n) が必ず有限の値を取ることの証明を与えよう。この証明の背後にある理論は、現在、WQO 理論という名の下で統一された理論である。その準備として、普遍代数における等式理論の概念を拡張した、不等式理論の概念を導入する。

3.1. 等式理論，不等式理論，自由代数

順序数の理論は、一変数多項式上の順序構造からスタートした。また、ヴェブレン階層までの順序数の理論もまた、いくつかの公理を満たす項の形式理論として導入した。様々な理論を、項の形式理論として取り扱うことによって、物事を統一的に見られることを明らかにしていこう。

等式理論の基本: 本稿で主に用いるものは、項の間の不等式の理論であるが、その前に項の間の等式理論を導入しておくことにする。等式理論は、普遍代数 (universal algebra) や数理論理学 (mathematical logic) と呼ばれる分野における基本概念である。

定義 3.1 (言語と項). 言語 (*language*) とは、引数が指定された関数記号 (*function symbols*) の集合である。引数 n の関数記号は、 n 変数関数記号と呼ばれ、特に 0-変数関数記号は定数記号 (*constant symbol*) と呼ばれる。言語 \mathcal{L} の項 (*term*) は、以下のように帰納的に定義されるものである。

1. 定数記号および変数記号は項である。
2. もし f が n 変数関数記号であり、 t_1, t_2, \dots, t_n が項ならば、 $f(t_1, t_2, \dots, t_n)$ は項である。

変数記号を含まない項は、閉項 (*closed term*) と呼ばれる。

例 3.2 (モノイドと群の言語). モノイドの言語 \mathcal{L}_{Mon} は 2 項関数記号 $*$ と定数記号 ε からなる集合 $\{*, \varepsilon\}$ で

ある．群の言語 \mathcal{L}_{Grp} は， $*$ と e に 1 変数関数記号 \bullet^{-1} を加えた集合 $\{*, \varepsilon, \bullet^{-1}\}$ である．通常， $*(x, y)$ を $x * y$ と略記し， $\bullet^{-1}(x)$ を x^{-1} と略記する．このとき， x, y, z, u, v が変数記号ならば，たとえば $(x * \varepsilon) * y$ は \mathcal{L}_{Mon} および \mathcal{L}_{Grp} の項であり， $(z * (u^{-1} * v)) * \varepsilon$ は \mathcal{L}_{Grp} の項であるが， \mathcal{L}_{Mon} の項ではない．また，これらは閉項ではないが，たとえば $\varepsilon * \varepsilon^{-1}$ は閉項である．

定義 3.3 (等式理論)．言語 \mathcal{L} の項全体の集合を $\text{Term}(\mathcal{L})$ と書く． $\text{Term}(\mathcal{L})$ 上の 2 項関係 T は等式理論 (equational theory) と呼ばれる．このとき，関係 $T \vdash s = t$ を T を拡張する最小の合同関係として定義する．つまり，関係 $T \vdash s = t$ は，以下のように帰納的に定義される．

$$\frac{(s, t) \in T}{T \vdash s = t} \text{ (公理)} \qquad \frac{}{T \vdash t = t} \text{ (反射律)}$$

$$\frac{T \vdash s = t}{T \vdash t = s} \text{ (対称律)} \qquad \frac{T \vdash s = t \quad T \vdash t = u}{T \vdash s = u} \text{ (推移律)}$$

もし x が変数記号で， f が n 変数関数記号ならば，

$$\frac{T \vdash s(x) = t(x)}{T \vdash s(u) = t(u)} \text{ (代入)} \qquad \frac{T \vdash s_1 = t_1, \dots, s_n = t_n}{T \vdash f(s_1, \dots, s_n) = f(t_1, \dots, t_n)} \text{ (関数適用)}$$

例 3.4 (モノイドの公理)．モノイドの公理 T_{Mon} は，項の等式 “ $(x * y) * z = x * (y * z)$ ” および “ $\varepsilon * x = x$ ” と “ $x * \varepsilon = x$ ” を公理に持つ理論である．群の公理 T_{Grp} には，さらに項の等式 “ $x * x^{-1} = \varepsilon$ ” および “ $x^{-1} * x = \varepsilon$ ” が含まれる．つまり，モノイドの公理 T_{Mon} と群の公理 T_{Grp} は等式理論である．

不等式理論： それでは，等式理論のアイデアを拡張して，不等式理論の概念を導入しよう．

定義 3.5 (不等式理論)．言語 \mathcal{L} の項全体の集合を $\text{Term}(\mathcal{L})$ と書き， $\text{Term}(\mathcal{L})$ 上の 2 項関係を不等式理論 (inequational theory) と呼ぶことにする．このとき，関係 $Q \vdash s \leq t$ を以下のように帰納的に定義する．

$$\frac{(s, t) \in Q}{Q \vdash s \leq t} \text{ (公理)} \qquad \frac{}{Q \vdash t \leq t} \text{ (反射律)} \qquad \frac{Q \vdash s \leq t \quad Q \vdash t \leq u}{Q \vdash s \leq u} \text{ (推移律)}$$

もし x が変数記号で， f が n 変数関数記号ならば，

$$\frac{Q \vdash s(x) \leq t(x)}{Q \vdash s(u) \leq t(u)} \text{ (代入)} \qquad \frac{Q \vdash s_1 \leq t_1, \dots, s_n \leq t_n}{Q \vdash f(s_1, \dots, s_n) \leq f(t_1, \dots, t_n)} \text{ (関数適用)}$$

不等式理論は不等式公理しか持たないが，等式 $s = t$ は $s \leq t$ かつ $t \leq s$ と表すことができる．したがって，不等式理論は等式理論よりも広い概念であると考えられる．たとえば，半群の公理や群の公理も不等式理論である．

定義 3.6. いま, \mathcal{L} 上の不等式理論 Q が与えられているとしよう. このとき, 与えられた擬順序 $A = (|A|, \leq_A)$ が Q -生成する擬順序 $Q[A]$ というものを定義しよう. まず, 各 $a \in |A|$ について定数記号 \check{a} を言語 \mathcal{L} に加え, 拡張した言語 $\mathcal{L}_A = \mathcal{L} \cup \{\check{a} : a \in |A|\}$ を考える. このとき, 擬順序 A は, \mathcal{L}_A の不等式理論とみなすことができる:

$$\text{Ax}_A := \{(\check{a}, \check{b}) : a \leq_A b\}.$$

$\text{Term}_{\text{cl}}(\mathcal{L}_A)$ を \mathcal{L}_A -閉項全体の集合とすると, この上の擬順序を次によって定義できる.

$$s \leq_{Q[A]} t \iff Q \cup \text{Ax}_A \vdash s \leq t.$$

このとき, 次の擬順序を A が Q -生成する擬順序と呼ぶ.

$$Q[A] := (\text{Term}_{\text{cl}}(\mathcal{L}_A), \leq_{Q[A]})$$

注意. 等式理論 T の場合は, この商構造を考えたものは, 通常は, A が生成する自由 T -代数と呼ばれる. しかし, 上概念の商を取ったものは, A が生成する自由 Q -半順序などと呼ぶのが妥当だろうか.

一旦, 不等式理論はこのように等式理論のアナロジーとして定義しておく. しかし, もう少し広範な擬順序を取り扱いこともあり, 特に, 不等式理論の代入と関数適用の規則については条件を緩めると便利なこともある. つまり, 不等式理論は, 等号付き論理の一種だと思って, 等式公理と不等式公理を別枠で考えるメリットもある.

不等式理論上で生成される擬順序の例: まずは, 言語 $\{*, \varepsilon\}$ における以下の項の等式公理および不等式公理を考えてみよう.

(結合律) “ $(x * y) * z = x * (y * z)$ ”;

(交換) “ $x * y \leq y * x$ ”; (弱化) “ $x \leq x * y$ ”, “ $y \leq x * y$ ”; (縮約) “ $a * a \leq a$ ”.

一番上の行は, つまり半群の公理であり, 等式公理であるから, 不等式公理として書き直せる. 下の行について, 論理学の構造規則とのアナロジーから交換・弱化・縮約という名称を付けている. ただし, 既知の順序と対応付けるために少し技巧的な修正を施したりするので, 論理学とのアナロジーに拘るのはもしかするとあまり適切でない可能性もある.

(a) 積順序: 結合律だけからなる不等式理論を Q_\emptyset と書くことにする. 擬順序 A が Q_\emptyset -生成する擬順序とは, 積順序 \leq_A^n の非交叉和である.

この文脈では, 項 $a_1 * a_2 * \dots * a_n$ は (a_1, a_2, \dots, a_n) と略記される. 擬順序 A の積順序の非交叉和 \leq_A^* とは, 項 $a = (a_1, a_2, \dots, a_n)$ と $b = (b_1, b_2, \dots, b_m)$ について, 以下によって定義されるものである.

$$a \leq_A^* b \iff (n = m) \text{ and } (\forall i \leq n) a_i \leq_A b_i.$$

つまり, 同じ長さだった場合のみ, 成分毎に順序比較をしていることになる. 以後, $Q \cup \text{Ax}_A$ のことも $Q[A]$ と略記することにすれば, 以下が成立している.

命題 3.7. 擬順序 A が Q_\emptyset -生成する擬順序は, A の積順序の非交叉和 \leq_A^* と等しい.

$$s \leq_A^* t \iff Q_\emptyset[A] \vdash s \leq t.$$

Proof. (\Rightarrow) 順方向については, 関数適用の推論規則から従う.

$$\frac{\frac{a_1 \leq_A b_1, \dots, a_n \leq_A b_n}{Q_\emptyset[A] \vdash a_1 \leq b_1, \dots, a_n \leq b_n} \text{ (公理)}}{Q_\emptyset[A] \vdash (a_1, \dots, a_n) \leq (b_1, \dots, b_n)} \text{ (関数適用)}$$

(\Leftarrow) 逆方向について, 各推論規則の上式を \leq_A^* に変えたものが成立しているならば下式も \leq_A^* に変えたものが成立することを示せば十分である. 公理と代入は自明であり, 反射律と推移律は \leq_A^* が擬順序であることから従う. 関数適用について確認すればよいが,

$$\frac{s \leq_A^* u \quad t \leq_A^* v}{(s, t) \leq_A^* (u, v)} \text{ (関数適用)}$$

が成立することは積順序の定義から明らかである. □

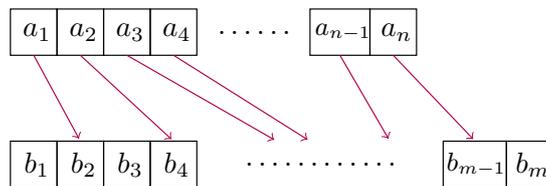
(b) 語の埋め込み順序: Q_\emptyset に弱化規則を加えた不等式理論を Q_w と書くことにする. 擬順序 A が Q_w -生成する擬順序とは, $|A|$ 上の語上の埋め込み順序である.

この文脈では, 項 $a_1 * a_2 * \dots * a_n$ の $*$ はしばしば略され, $a_1 a_2 \dots a_n$ と書かれ, アルファベット $|A|$ 上の語と呼ばれる. 語上の埋め込み順序 (*embeddability ordering*) とは, 語 $a = a_1 a_2 \dots a_n$ と $b = b_1 b_2 \dots b_m$ について, 以下によって定義されるものである.

$$a \leq_A^w b \iff \text{ある単射単調写像 } h: \{1, \dots, n\} \rightarrow \{1, \dots, m\} \text{ が存在して,}$$

$$\text{任意の } i \leq n \text{ について, } a_i \leq_A b_{h(i)} \text{ が成立する.}$$

ここで, h が単調であるとは, $i \leq j$ ならば $h(i) \leq h(j)$ であることを意味する.



これを $|A|$ -ラベル付き有限全順序上の埋め込み順序だと考えれば, 定義 1.8 と等しい概念であることがわかる.

命題 3.8. 擬順序 A が Q_w -生成する擬順序は, A 上の語上の埋め込み順序 \leq_A^w と等しい.

$$s \leq_A^w t \iff Q_w[A] \vdash s \leq t.$$

Proof. (\Rightarrow) 順方向について、単射単調写像 h が与えられているとする。単射性と単調性より、 $i < j$ ならば $h(i) < h(j)$ なので、 $b_{h(i-1)+1} \dots b_{h(i)-1} b_{h(i)}$ は空語ではない。ここで、 $h(-1) = 0$ とする。よって、仮定 $a_i \leq_A b_{h(i)}$ と弱化規則より、

$$\frac{\frac{a_i \leq_A b_{h(i)}}{Q_w[A] \vdash a_i \leq b_{h(i)}} \text{ (公理)}}{Q_w[A] \vdash a_i \leq b_{h(i-1)+1} \dots b_{h(i)}} \frac{Q_w[A] \vdash b_{h(i)} \leq b_{h(i-1)+1} \dots b_{h(i)}}{\text{ (推移律)}} \text{ (弱化)}$$

を得る。これに関数適用の推論規則を用いれば、以下を得る。

$$\frac{Q_w[A] \vdash a_1 \leq b_1 \dots b_{h(1)} \quad \dots \quad a_n \leq b_{h(n-1)+1} \dots b_{h(n)}}{Q_\emptyset[A] \vdash a_1 \dots a_n \leq b_1 \dots b_{h(n)}} \text{ (関数適用)}$$

$$\frac{Q_\emptyset[A] \vdash a_1 \dots a_n \leq b_1 \dots b_{h(n)}}{Q_\emptyset[A] \vdash a_1 \dots a_n \leq b_1 \dots b_m} \text{ (弱化)}$$

(\Leftarrow) まず、埋め込み順序が弱化規則を満たすことは容易に確認できる。具体的には、 $s \leq_A^w st$ は h を恒等写像とすればよく、 $s = a_1 \dots a_n$ の場合、 $t \leq_A^w st$ を示すには $h(i) = n + i$ を考えればよい。関数適用についても、 $h_1: s \leq_A^w u$ と $h_2: t \leq_A^w v$ を仮定すると、 $s = a_1 \dots a_n$ かつ $u = b_1 \dots b_m$ であったとすれば、

$$h(i) = \begin{cases} h_1(i) & \text{if } i \leq n \\ m + h_2(i - n) & \text{if } i > n \end{cases}$$

とすれば、 $h: st \leq_A^w uv$ であることは明らかである。 \square

(c) 語の準同型順序: Q_\emptyset に弱化規則と縮約規則を加えた不等式理論を Q_{wc} と書くことにする。擬順序 A が Q_{wc} -生成する擬順序とは、 $|A|$ 上の語上の準同型順序である。

ただし、ここで交換規則がない場合の縮約規則については、定数記号 a にしか適用できないとする。つまり、各不等式規則の代入規則を除去して、以下の規則を扱うことにする。

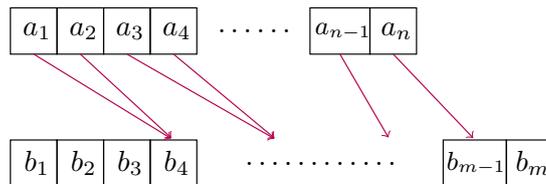
$$\frac{}{Q \vdash s * t \leq s * a * t} \text{ (弱化)} \qquad \frac{}{Q \vdash s * a * a * t \leq s * a * t} \text{ (縮約)}$$

ここで、 s, t は項であるが、 a, b は定数記号であるとする。語上の準同型順序 (*homomorphic ordering*) とは、語 $a = a_1 a_2 \dots a_n$ と $b = b_1 b_2 \dots b_m$ について、以下によって定義されるものである。

$$a \leq_A^{wc} b \iff \text{ある単調写像 } h: \{1, \dots, n\} \rightarrow \{1, \dots, m\} \text{ が存在して、}$$

$$\text{任意の } i \leq n \text{ について、} a_i \leq_A b_{h(i)} \text{ が成立する。}$$

埋め込み順序との違いは、単射性を要求しないことである。



命題 3.9. 擬順序 A が Q_{wc} -生成する擬順序は, A 上の語上の準同型順序 \leq_A^{wc} と等しい.

$$s \leq_A^{wc} t \iff Q_{wc}[A] \vdash s \leq t.$$

Proof. (\Rightarrow) 順方向について, 単調写像 h が与えられているとする. 命題 3.8 のときとは異なり, 単射性を要求していないため, $h(i-1) = h(i)$ のときに, $\beta_i := b_{h(i-1)+1} \dots b_{h(i)-1} b_{h(i)}$ が空語になる. もし β_i が空語ならば $\beta'_i = b_{h(i)}$ とし, さもなくば $\beta'_i = \beta_i$ とする. このとき, β'_i の最後の文字は必ず $b_{h(i)}$ であるから, 仮定 $a_i \leq_A b_{h(i)}$ と弱化規則より,

$$\frac{a_i \leq_A b_{h(i)} \quad (公理)}{Q_w[A] \vdash a_i \leq b_{h(i)}} \quad \frac{Q_w[A] \vdash b_{h(i)} \leq \beta'_i \quad (弱化)}{Q_w[A] \vdash a_i \leq \beta'_i} \quad (推移律)$$

を得る. これに関数適用の推論規則を用いれば, 以下を得る.

$$\frac{Q_w[A] \vdash a_1 \leq \beta'_1 \quad \dots \quad a_n \leq \beta'_n \quad (関数適用)}{Q_w[A] \vdash a_1 \dots a_n \leq \beta'_1 \dots \beta'_n}$$

もし β_i が空語ならば, $h(i-1) = h(i)$ であるから, β'_{i-1} と β'_i の最後の文字は $b_{h(i-1)} = b_{h(i)}$ となる. よって, β'_{i-1} から最後の文字を取り除いたものを β'^{-}_{i-1} と書けば, 縮約規則によって,

$$Q_w[A] \vdash \beta'^{-}_{i-1} \beta'_i = \beta'^{-}_{i-1} b_{h(i-1)} b_{h(i)} = \beta'^{-}_{i-1} b_{h(i-1)} b_{h(i-1)} \leq \beta'^{-}_{i-1} b_{h(i-1)} = \beta'^{-}_{i-1}$$

を得る. つまり,

$$\beta_i = \emptyset \implies Q_w[A] \vdash \beta'_1 \dots \beta'^{-}_{i-1} \beta'_i \beta'_{i+1} \dots \beta'_n \leq \beta'_1 \dots \beta'^{-}_{i-1} \beta'_{i+1} \dots \beta'_n$$

のようにして, β'_i を取り除くことができる. しかし, β_i が空語でないならば $\beta'_i = \beta_i$ であったから, これは

$$Q_w[A] \vdash \beta'_1 \dots \beta'_n \leq \beta_1 \dots \beta_n$$

を導く. したがって, 推移律と弱化規則より, 目的の結論が得られる.

$$\frac{Q_w[A] \vdash a_1 \dots a_n \leq \beta'_1 \dots \beta'_n \quad Q_w[A] \vdash \beta'_1 \dots \beta'_n \leq \beta_1 \dots \beta_n \quad (推移律)}{Q_w[A] \vdash a_1 \dots a_n \leq \beta_1 \dots \beta_n = b_1 \dots b_{h(n)}} \quad (弱化)$$

$$\frac{Q_w[A] \vdash a_1 \dots a_n \leq b_1 \dots b_m \quad (弱化)}{Q_w[A] \vdash a_1 \dots a_n \leq b_1 \dots b_m}$$

(\Leftarrow) 準同型順序が弱化規則と関数適用規則を満たすことは, 命題 3.8 の証明と同様である. 縮約規則 $s * a * a * t \leq_A^{wc} s * a * t$ も自明である. 具体的には, $s = s_1 * \dots * s_n$ かつ $t = t_1 * \dots * t_m$ とすれば, $i \leq n+1$ について $h(i) = i$ とし, $i > n+1$ について $h(i) = i-1$ とすれば, $s * a * a * t \leq_A^{wc} s * a * t$ が導かれる. \square

(d) 有限集合の支配順序: Q_\emptyset に交換, 弱化, 縮約規則を加えた不等式理論を Q_{ewc} と書くことにする. 擬順序 A が Q_{ewc} -生成する擬順序とは, $|A|$ の有限部分集合上の支配順序である.

以下, 記号 $*$ を略記するが, 交換規則がある場合, たとえば項 $abbaaccabc$ は常に $aaaabbbccc$ のように整理できる. この上で縮約規則を用いると, すべての重複を削除して, abc という形にでき

る．この文脈では， $a_1 a_2 \dots a_n$ は $\{a_1, a_2, \dots, a_n\}$ と略記される．上で述べた項の例をこの記法で書き直せば，

$$\{a, b, b, a, a, c, c, a, b, c\} = \{a, a, a, b, b, b, c, c, c\} = \{a, b, c\}$$

ということである．以上の議論より，与えられた A について，交換と縮約がある場合の \mathcal{L}_A -閉項は， $|A|$ の有限部分集合と同一視できる．

$|A|$ の部分集合 $U, V \subseteq |A|$ について，支配順序 (*domination order*) は以下のように定義される．

$$U \leq_A^{\text{ewc}} V \iff (\forall x \in U)(\exists y \in V) x \leq_A y.$$

命題 3.10. 擬順序 A が Q_{ewc} -生成する擬順序は， A の有限部分集合上の支配順序 \leq_A^{ewc} と等しい．

$$s \leq_A^{\text{ewc}} t \iff Q_{\text{ewc}}[A] \vdash s \leq t.$$

Proof. (\Rightarrow) 項 $s = \{a_1, \dots, a_n\}$ と $t = \{b_1, \dots, b_m\}$ について， $s \leq_A^{\text{ewc}} t$ を仮定する．特に，ある関数 $h: \{1, \dots, n\} \rightarrow \{1, \dots, m\}$ が存在して，任意の $i \leq n$ について $a_i \leq_A b_{h(i)}$ が成立する．したがって，演算の関数適用によって，

$$\{a_1, a_2, \dots, a_n\} \leq \{b_{h(1)}, b_{h(2)}, \dots, b_{h(n)}\}$$

が成立する．しかし， $t' = \{b_{h(1)}, \dots, b_{h(n)}\}$ は $t = \{b_1, \dots, b_m\}$ の部分集合であるから，交換規則を用いて t' に現れる定数記号を順番通りに並び替え， t' の足りない要素を弱化規則で増やし， t' の重複した要素を縮約規則を消去することによって，目的の式は示される．

$$s = \{a_1, a_2, \dots, a_n\} \leq \{b_{h(1)}, b_{h(2)}, \dots, b_{h(n)}\} \leq \{b_1, b_2, \dots, b_m\} = t.$$

(\Leftarrow) 証明は容易であるので省略する． □

(e) 一変数多項式上の順序: 不等式理論を Q_{ewc} の縮約規則を少し弱めると，少し興味深い不等式理論を生み出すことができる．いま，縮約規則を少し弱めた公理を考えてみよう．

$$\text{(弱い縮約)} \quad a < b \implies a * a < b$$

このとき， Q_{\emptyset} に交換，弱化，弱い縮約規則を加えた不等式理論を Q'_{ewc} と書くことにする．この場合も交換規則があるので，項 $abbaaccabc$ は常に $aaaabbbccc$ と整理できるが，縮約が弱いため，定数記号の重複を削除できるとは限らない．このため，この項を $a^4 b^3 c^3$ と多変数単項式のように記述するのは妥当であろう．しかし，実際には次のいずれかの記法を使うと，順序の意味を理解しやすい．

$$4x^a + 3x^b + 3x^c,$$

$$\omega^a \cdot 4 + \omega^b \cdot 3 + \omega^c \cdot 3.$$

ここで， x や ω は形式的なダミー記号であり，特に何の意味ももたない．この不等式理論 Q'_{ewc} は，多重集合上の擬順序という形で紹介されることもある．

自然数上の標準的な順序を ω と書けば， ω から Q'_{ewc} -生成される順序は例 2.12 で紹介した 1 変数多項式上の順序と同一であり，この順序型は ω^ω と書かれる．続いて ω^ω から Q'_{ewc} -生成される

順序の順序型は ω^{ω} である．一般に，順序型 α の整列順序から Q'_{ewc} -生成される順序の順序型は ω^α である．

3.2. 木と森の不等式理論

ここまでで $Q_w[A]$ は A -ラベル付き全順序の埋め込み順序であることを見た．これらの発想を拡張すれば， A -ラベル付き木の \wedge -埋め込み順序もまた不等式理論の文脈で説明できるだろうか．そのヒントは既に第 1.4 節で紹介した，木の項表示である．木を代数的に取り扱うためには，2 種類の 2 項演算，すなわち「和」と「積」の両方が必要である．つまり，ここからは，言語 $\{+, *\}$ を考える．和と積に関する最低限の公理として，以下を要求する．

定義 3.11. 言語 $\{+, *\}$ の等式理論 R は，以下からなるものである．

(結合律) “ $(x * y) * z = x * (y * z)$ ”; “ $(x + y) + z = x + (y + z)$ ”;

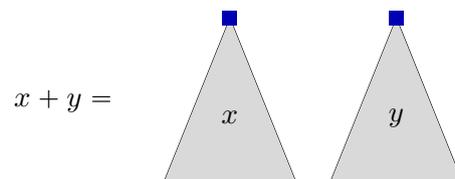
(和の交換規則) “ $x + y = y + x$ ”; (右分配律) “ $(x + y) * z = x * z + y * z$ ”.

等式理論 R の項によって，森を表すことができる．ここで，各連結成分が木となっているようなグラフのことを森 (*forest*) と呼ぶ．ただし，本稿における木とは根付き木であったから，連結成分ごとに根があることに注意する．同値であるが，森とは，0 個以上の木の非交叉和である，と言ってもよい．また，集合 L に対して， L -ラベル付き森 (L -labeled forest) とは，森 T と写像 $\ell_T: T \rightarrow L$ の対である．そして，擬順序 A が与えられているとき， $R[A]$ における項は， A -ラベル付き森に対応する．

まず，各項がいかなるラベル付き森を表しているかについて解説しよう．定数記号 a は， a でラベルが付けられた唯一の頂点からなる木である．

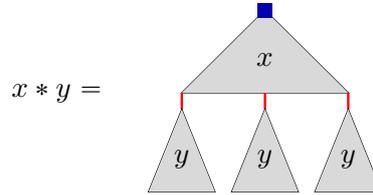
$$a = \text{● } a$$

和 $x + y$ は，ラベル付き森 x, y の非交叉和を表す．つまり，森をグラフだと思えば，グラフ x と y が離れた所へ配置され， x と y を繋ぐ辺は存在しない．

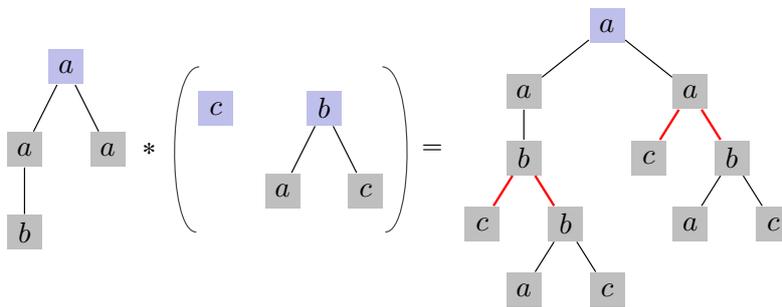


積 $x * y$ は，ラベル付き森 x の各葉に森 y のコピーのすべての根を接続する辺を加えたものを意

味する．たとえば， x と y が共に木の場合には，次のようなものである．



ただし， y が森の場合には， x の各葉毎に y の根の数だけ接続辺を加える必要がある．具体例で説明すれば，以下のような感じである．



左図は，以下のように項表示することができる．

$$s = (a * (a * b + a)) * (c + b * (a + c))$$

しかし，右図は，第 1.4 節で紹介した項表示をそのまま適用すれば，次のようになる．

$$t = a * \left[(a * b * (c + b * (a + c))) + a * (c + b * (a + c)) \right]$$

この 2 つの項の同値性は，右分配律によって保証されている．以下， $*$ は省略して書いている．

$$s = a(ab + a)(c + b(a + c)) = a[ab(c + b(a + c)) + a(c + b(a + c))] = t.$$

この議論を一般的に説明しよう．まず，項として 2 種類のものがあることに注目する．つまり，木を表す項と木にならない項，あるいは連結項と不連結項と言ってもよい．項としての側面を強調すれば，最も外の演算が $+$ 以外の項と $+$ の項である．不連結項は， $+$ によって生み出され， $*$ からは生み出されない．

定義 3.12. 不等式理論 R において，以下の形の項を標準形と呼ぶ．

$$t = a * (u_1 + u_2 + \cdots + u_\ell) \quad \text{or} \quad t = u_1 + u_2 + \cdots + u_\ell.$$

ここで， a は定数記号であり，各 u_k は連結項であり，右の場合には $\ell(i) \geq 2$ であるとする．ただし，項 t_i が連結項ならば左の形であり，不連結項ならば右の形であるとする．

主張．不等式理論 R において，任意の項は標準形と等しい．

Proof. 帰納法によって示す．まず， s, t が標準形ならば $s + t$ も標準形であることは容易に確認できる．しかし， s, t が標準形であっても $s * t$ が標準形であるとは限らないが，たとえば， $s = a * (u_1 + \dots + u_\ell)$ だったとしよう．以下， $*$ は省略して記すが，右分配律によって以下を示せる．

$$\begin{aligned} st &= a(u_1 + \dots + u_\ell)t = a((u_1 + \dots + u_{\ell-1}) + u_\ell)t \\ &= a((u_1 + \dots + u_{\ell-1})t + u_\ell t) = \dots = a(u_1 t + \dots + u_\ell t). \end{aligned}$$

各 $u_i * t$ の最も外の演算は $*$ であるから， $u_i * t$ は連結項であり，したがって，この右下の式は標準形である．よって，項の構成に関する帰納法によって，任意の項は標準形と等しいことが示された． \square

以上より， $*$ 単体だと語の理論であったが， $+$ と $*$ を組み合わせると森の理論を生み出すことができる．

(f) 木の準同型順序: この不等式理論 R に対して，新たに和と積に関する弱化規則

$$\text{(弱化)} \quad "x \leq x + y"; \quad "x \leq x * y"; \quad "x \leq y * x".$$

を加えた不等式理論 R_w を考えていこう．一応，注意しておくとして，和については可換性があるので，弱化規則の式は一つ考えれば十分である．

さて，実は R_w の分析はやや難しいので，先に縮約規則を加えた不等式理論 R_{wc} を分析することにする．ただし，ここでは和と積に関する縮約規則を加えるが，考える縮約規則は以下のものである．

$$"s * a * a * t \leq s * a * t"; \quad "s + u + u + t \leq s + u + t".$$

ここで， s, t, u は項であり， a は定数記号であるとする．つまり， $*$ に関する縮約規則は定数記号にしか適用できないが， $+$ に関する縮約規則は任意の項に適用できる． R_w にこの規則を加えた理論を R_{wc} と書くことにする．不等式理論 Q_{wc} が語の準同型順序に対応していたように，不等式理論 R_{wc} に対応する順序は，森の準同型順序である．写像 $h: S \rightarrow T$ が $|A|$ -ラベル付き森 S から T への準同型 (*homomorphism*) とは，以下の条件を満たすことである．

$$\begin{aligned} \sigma \sqsubseteq \tau &\implies h(\sigma) \sqsubseteq h(\tau), \\ \ell_S(\sigma) &\leq_A \ell_T(h(\sigma)). \end{aligned}$$

つまり，ラベル増大な単調写像である． S から T への準同型が存在するとき， $S \leq_A^{wc} T$ と書くことにする．

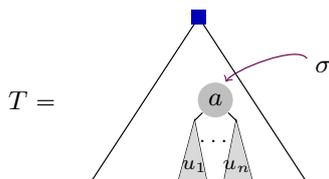
命題 3.13. 擬順序 A が R_{wc} -生成する擬順序は， $|A|$ -ラベル付き森の準同型順序 \leq_A^{wc} と等しい．

$$s \leq_A^{wc} t \iff R_{wc}[A] \vdash s \leq t.$$

Proof. (\Rightarrow) 準同型 $h: S \rightarrow T$ が与えられていると仮定する．まずは，記法の準備から始める．森 T のノード σ をひとつ固定したとき， σ の先のノード全体 $T_\sigma := \{\tau \in T : \sigma \sqsubseteq \tau\}$ は， T を表す項 t_T の部分項 t_σ に対応する．いま

$$t_\sigma = a * (u_1 + u_2 + \cdots + u_n)$$

の形であると仮定できる．したがって，特に $l_T(\sigma) = a$ である．



また， t_σ^+ を σ を真に拡張するノード全体を表す項，つまり $t_\sigma^+ = u_1 + \cdots + u_n$ とする． S 側のノード σ についても，同様にして項 s_σ と s_σ^+ を与える．

それでは，証明に入ろう．まず， S の各ノード σ の子ノードを $\sigma_1, \dots, \sigma_k$ とリストすれば，ノード σ に対する項 s_σ は次のように表すことができる．

$$s_\sigma = l_S(\sigma) * (s_{\sigma_1} + \cdots + s_{\sigma_k}).$$

帰納的に $s_{\sigma_j} \leq t_{h(\sigma_j)}$ を仮定する．このとき， $h(\sigma) \sqsubseteq h(\sigma_j)$ であることから， $h(\sigma)$ は $h(\sigma_j)$ を部分項として含むので，弱化規則を繰り返し用いることにより，

$$s_{\sigma_j} \leq t_{h(\sigma_j)} \leq t_{h(\sigma)}$$

を得る．したがって， $+$ に関する縮約規則を用いれば，

$$s_\sigma^+ = \sum_{j=1}^k s_{\sigma_j} \leq t_{h(\sigma)} + \cdots + t_{h(\sigma)} \leq t_{h(\sigma)}$$

を導くことができる．よって，仮定 $l_S(\sigma) \leq l_T(h(\sigma))$ と $*$ に関する関数適用と縮約規則を用いれば，以下を得る．

$$\begin{aligned} s_\sigma &= l_S(\sigma) * \sum_{j=1}^k s_{\sigma_j} \leq l_T(h(\sigma)) * t_{h(\sigma)} \\ &= l_T(h(\sigma)) * l_T(h(\sigma)) * t_{h(\sigma)}^+ \leq l_T(h(\sigma)) * t_{h(\sigma)}^+ = t_{h(\sigma)}. \end{aligned}$$

(\Leftarrow) 項の木表示の定義より，分配則は自明である． $+$ の交換規則は，木の分岐の左右入れ替えを意味するが，これは準同型順序で同一視できる．また， $+$ の弱化について， $s \leq_A^w s + t$ は t を無視して，行き先も s 側に移せばよいので自明である． $*$ の弱化 $s \leq_A^w s * t$ と $s \leq_A^w t * s$ も同様である．関数適用についても，たとえば $s \leq_A^w u$ と $t \leq_A^w v$ に対する準同型を組み合わせれば， $s * t \leq_A^w u * v$ および $s + t \leq_A^w u + v$ に対する準同型を作ることができる．縮約についても容易である． \square

(g) 木の \perp -埋め込み順序: それでは，森の不等式理論 R に対して，和と積に関する弱化規則

$$(\text{弱化}) \quad "x \leq x + y"; \quad "x \leq x * y"; \quad "x \leq y * x".$$

を加えた不等式理論 R_w を考えていこう．前に注意したように，和については可換性があるので，弱化規則の式は一つ考えれば十分である．

不等式理論 Q_w が語の埋め込み順序に対応していたことから、 R_w は森の埋め込み順序に対応していると予想できる。しかし、残念ながら、これは \wedge -埋め込み順序よりも少し強くなってしまふ。具体的には、 R_w は分岐を保つ埋め込みに対応する。これを正確に述べるために、森 T のノード σ, τ について、以下の記法を用いる。

$$\sigma \perp \tau \iff \sigma \sqsubset \tau \text{ and } \tau \sqsubset \sigma.$$

このとき、 $h: S \rightarrow T$ が $|A|$ -ラベル付き森 S から T への分岐を保つ埋め込み、あるいは略して \perp -埋め込みであるとは、以下の条件を満たすことである。

$$\begin{aligned} \sigma \sqsubset \tau &\implies h(\sigma) \sqsubset h(\tau), \\ \sigma \not\sqsubset \tau &\implies h(\sigma) \not\sqsubset h(\tau), \\ \sigma \perp \tau &\implies h(\sigma) \perp h(\tau), \\ \ell_S(\sigma) &\leq_A \ell_T(h(\sigma)). \end{aligned}$$

つまり、上から順に、単調性、単射性、分岐保存性、ラベル増大性である。 S から T への \perp -埋め込みが存在するとき、 $S \leq_A^w T$ と書くことにする。

命題 3.14. 擬順序 A が R_w -生成する擬順序は、 $|A|$ -ラベル付き森の \perp -埋め込み順序 \leq_A^w と等しい。

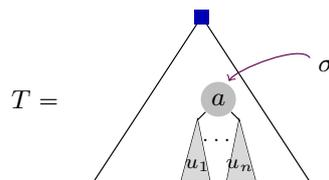
$$s \leq_A^w t \iff R_w[A] \vdash s \leq t.$$

Proof. (\implies) \perp -埋め込み $h: S \rightarrow T$ が与えられていると仮定する。基本的なアイデアとして、まず、 S に対応する項 s と埋め込み h の像である T の部分木に対応する項 t^* に対して、関数適用によって $s \leq t^*$ は保証される。さらに、 T 全体に対応する項を t とすると、 $+$ と $*$ に関する弱化規則を用いて $t^* \leq t$ を導くことができる。あとは推移律より $s \leq t$ が従う。

このように直感的には明らかであるが、形式的に書き下そうとすると、記法の点で厄介である。まず、記法の整理から始めよう。森 T のノード σ をひとつ固定したとき、 σ の先のノード全体 $T_\sigma := \{\tau \in T : \sigma \sqsubset \tau\}$ は、 T を表す項 t_T の部分項 t_σ に対応する。いま

$$t_\sigma = a * (u_1 + u_2 + \cdots + u_n)$$

の形であると仮定できる。したがって、特に $\ell_T(\sigma) = a$ である。



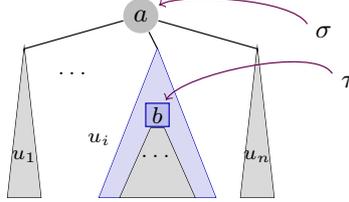
もし $\sigma \sqsubset \tau$ であれば、 τ に対応する項 t_τ はいずれかの u_i の部分項になっている。つまり、ある項 $v_i(x)$ が存在して、 $u_i = v_i(t_\tau)$ となる。先ほどと同様に、

$$t_\tau = b * (v_1 + v_2 + \cdots + v_m)$$

の形である．このとき，

$$t_{(\sigma, \tau \downarrow)}(x) := v_i(b * x), \quad t_{(\sigma, \tau \downarrow)} := t_{(\sigma, \tau \downarrow)}(\emptyset) = v_i(b)$$

として定義する．ここで， t_τ^+ を τ を真に拡張するノード全体を表す項，つまり $t_\tau^+ = v_1 + \dots + v_n$ とすれば，もちろん $t_{(\sigma, \tau \downarrow)}(t_\tau^+) = u_i$ である．図で表せば， $t_{(\sigma, \tau \downarrow)}$ とは，以下の図中で青色でハイライトした部分である．



それでは，証明に入ろう．議論を簡易化するために， $h: S \rightarrow T$ は \wedge -埋め込みであるとする．まず， S の各ノード σ の子ノードを $\sigma_1, \dots, \sigma_k$ とリストすれば，ノード σ に対する項 s_σ は次のように表すことができる．

$$s_\sigma = \ell_S(\sigma) * (\ell_S(\sigma_1) * s_{\sigma_1}^+ + \dots + \ell_S(\sigma_k) * s_{\sigma_k}^+).$$

いま， σ_j を真に拡張するノードの埋め込み h による行き先は $h(\sigma_j)$ を真に拡張するので，帰納的に $s_{\sigma_j}^+ \leq t_{h(\sigma_j)}^+$ を仮定できる．各 $j \leq k$ について，項 $t_{(h(\sigma), h(\sigma_j) \downarrow)}(x)$ は項 $\ell_T(h(\sigma_j)) * x$ を部分項として含むので，弱化規則を繰り返し用いて，次を得る．

$$\ell_T(h(\sigma_j)) * x \leq t_{(h(\sigma), h(\sigma_j) \downarrow)}(x). \quad (3)$$

したがって，帰納的仮定，関数適用規則，代入規則，推移律を用いて，

$$s_{\sigma_j} = \ell_S(\sigma_j) * s_{\sigma_j}^+ \leq \ell_T(h(\sigma_j)) * t_{h(\sigma_j)}^+ \leq t_{(h(\sigma), h(\sigma_j) \downarrow)}(t_{h(\sigma_j)}^+) \quad (4)$$

を導ける．次に， T のノード $h(\sigma)$ の子ノードを τ_1, \dots, τ_ℓ とリストし， $n(j)$ を $h(\sigma) \sqsubset \tau_{n(j)} \sqsubseteq h(\sigma_j)$ となる $h(\sigma)$ の唯一のものとする．+ に関する関数適用規則と弱化規則より，以下を得る．

$$s_\sigma^+ = \sum_{j=1}^k (\ell_S(\sigma_j) * s_{\sigma_j}^+) \leq \sum_{j=1}^k t_{(h(\sigma), h(\sigma_j) \downarrow)}(t_{h(\sigma_j)}^+) = \sum_{j=1}^k t_{\tau_{n(j)}} \leq \sum_{n=1}^{\ell} t_{\tau_n} = t_{h(\sigma)}^+.$$

さらに，埋め込みの性質 $\ell_S(\sigma) \leq \ell_T(h(\sigma))$ と * に関する関数適用より， $s_\sigma^+ \leq t_{h(\sigma)}^+$ から $s_\sigma \leq t_{h(\sigma)}$ が導かれる．したがって，帰納法によって主張は導かれた．

$h: S \rightarrow T$ が \perp -埋め込みである場合を考えよう．上の議論では， $s_{\sigma_j} \leq t_{\tau_{n(j)}}$ であることを示していたが， \perp -埋め込みの場合には， $i \neq j$ であっても $n(i) = n(j)$ になってしまうことがあり得る．このため，直後の子ノードで分解するのではなく， $h(\sigma_i)$ と $h(\sigma_j)$ の分岐が発生した部分で分解を行う．たとえば， σ の子ノードが σ_1 と σ_2 であり， $h(\sigma_1)$ と $h(\sigma_2)$ の分岐が $h(\sigma)$ の直後でないノード τ_{12} である場合を考えよう．このとき，前と同様の議論によって以下を得る．

$$\begin{aligned} s_\sigma^+ &= s_{\sigma_1} + s_{\sigma_2} \leq t_{(\tau_{12}, h(\sigma_1) \downarrow)}(t_{h(\sigma_1)}^+) + t_{(\tau_{12}, h(\sigma_2) \downarrow)}(t_{h(\sigma_2)}^+) \\ &\leq \tau_{12}^+ \leq t_{(h(\sigma), \tau_{12} \downarrow)}(\tau_{12}^+) \leq t_{h(\sigma)}^+. \end{aligned} \quad (5)$$

一般の場合も議論は同様である．

(\Leftarrow) 項の木表示の定義より，分配則は自明である．+ の交換規則は，木の分岐の左右入れ替えを意味するが，これは \perp -埋め込み順序で同一視できる．また，+ の弱化について， $s \leq_A^w s + t$ は t を無視して s に埋め込めばよいので自明である．* の弱化 $s \leq_A^w s * t$ と $s \leq_A^w t * s$ も同様である．関数適用についても，たとえば $s \leq_A^w u$ と $t \leq_A^w v$ に対する \perp -埋め込みを組み合わせれば， $s * t \leq_A^w u * v$ および $s + t \leq_A^w u + v$ に対する \wedge -埋め込みを作ることができる． \square

(h) 木の \wedge -埋め込み順序: 先ほどの不等式理論 R_w を修正して, \wedge -埋め込み順序を表すことはできないだろうか. その方法として, 項として 2 種類のものがあったことを思い出そう. つまり, 木を表す項 (連結項) と木にならない項 (不連結項) である. 不連結項は, $+$ によって生み出され, $*$ からは生み出されない. したがって, 「 $*$ に関する規則は連結項にのみ適用できる」という制限を課するのは妥当だと思われる. ただし, $+$ に関する規則は如何なる項に適用してもよい. しかし, そうすると $*$ は不連結項について何も言えなくなってしまうため, 以下の弱めた関数適用規則は認めるものとする.

$$\frac{a \leq b \quad s \leq t}{a * s \leq b * t} \text{ (弱い関数適用)}$$

ここで, a, b は定数記号であり, s, t は任意の項である. このように修正した不等式理論を R_w^- と書くことにしよう.

さて, 現在扱っているのは森なので, 森に関する \wedge -埋め込み順序の正確な定義を与えておこう. いま, ノード σ と τ が同じ連結成分に属していることを $\sigma \sim \tau$ と書くことにする. このとき, $h: S \rightarrow T$ が $|A|$ -ラベル付き森 S から T への \wedge -埋め込みみであるとは, 以下の条件を満たすことである.

$$\begin{aligned} \sigma \sqsubseteq \tau &\implies h(\sigma) \sqsubseteq h(\tau), \\ \sigma \not\sqsubseteq \tau &\implies h(\sigma) \not\sqsubseteq h(\tau), \\ \sigma \not\sim \tau &\implies h(\sigma) \not\sim h(\tau), \\ h(\sigma \wedge \tau) &= h(\sigma) \wedge h(\tau), \\ \ell_S(\sigma) &\leq_A \ell_T(h(\sigma)). \end{aligned}$$

木の \wedge -埋め込みの違いは, 3 行目の性質で, 異なる連結成分に属すノードは異なる連結成分に移す, というものである. S から T への \wedge -埋め込みが存在するとき, $S \leq_A^{w-} T$ と書くことにする.

命題 3.15. 擬順序 A が R_w^- -生成する擬順序は, $|A|$ -ラベル付き森の \wedge -埋め込み順序 \leq_A^{w-} と等しい.

$$s \leq_A^{w-} t \iff R_w^-[A] \vdash s \leq t.$$

Proof. (\implies) 順方向については, 命題 3.14 の証明の $*$ に関する規則を用いている部分を注意深く分析すればよい. 式 (3) は単独の定数記号を別の項に埋め込む操作なので, R_w^- で問題なく実行できる. たとえば, $a = \ell_S(\sigma) \leq \ell_T(h(\sigma)) = a'$ だったとして, 項 a が項 $c * (a' + b)$ に埋め込まれている場合には, 連結項に対する積 $*$ の弱化, 和の弱化, 弱い関数適用を用いて, 以下を得る.

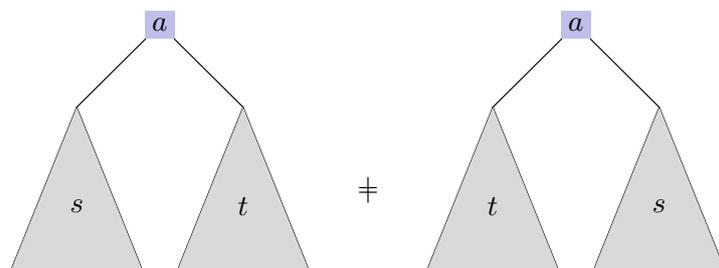
$$\frac{\frac{\frac{a' \leq a'}{a' \leq c * a'} \text{ (* 弱化)} \quad \frac{\frac{a' \leq a'}{a' \leq a' + b} \text{ (+ 弱化)}}{c * a' \leq c * (a' + b)} \text{ (弱い関数適用)}}{a' \leq c * (a' + b)} \text{ (推移律)}}{a \leq c + (a' + b)} \text{ (推移律)}$$

命題 3.14 の証明のその次の式 (4) における $*$ の関数適用は, まさに弱い関数適用である. 同様に, $\ell_S(\sigma) \leq \ell_T(h(\sigma))$ と $s_\sigma^+ \leq t_{h(\sigma)}^+$ から $s_\sigma \leq t_{h(\sigma)}$ を導く最後のステップも弱い関数適用である.

一応、 \perp -埋め込みについては、この議論が通用しないことを説明しておこう。命題 3.14 の証明の式 (5) において、一行目の不等式までは示せるのだが、そこに現れる不連結項 τ_{12}^+ に $t_{(h(\sigma), \tau_{12}^+)}$ を被せて全体の不等式を示すには、不連結項に関する $*$ の弱化規則が必要である。しかし、 R_w^- は不連結項に関する $*$ の弱化規則を持たないため、式 (5) を示せないのである。

(\Leftarrow) $+$ の交換法則、弱化規則、関数適用については自明である。 $*$ の弱化について、 $s \leq_A^{w-} s * t$ は自明であるが、 $s \leq_A^{w-} t * s$ については、 s が連結項に制限されていることから従う。弱い関数適用について、 $a \leq_A^{w-} b$ および $s \leq_A^{w-} t$ を仮定する。 s, t が不連結項の場合のみ考えると、 \leq_A^{w-} の定義より、 s の異なる連結成分のノードは t の異なる連結成分のノードに移されている。したがって、 $a \leq_A^{w-} b$ に対応する \wedge -埋め込みと $s \leq_A^{w-} t$ に対応する \wedge -埋め込みを単純に貼り合わせれば、 \wedge -埋め込みになっている。□

(i) 木の左右を保つ \wedge -埋め込み順序: さて、不等式理論 R は $+$ は交換法則を持つということを仮定していたが、和の交換法則を取り除いても問題ない。その場合、 $s + t \neq t + s$ になり得るので、項を森として見たときには、分岐の順番まで考慮に入れた、順序付き森と呼ばれるものに相当する。



このような理論に $+$ に関する弱化規則を加え、さらに (f) のように、 $*$ の連結項に対する弱化規則と弱い関数適用を加えた不等式理論を R_w° と書くことにする。このとき、不等式理論 R_w° から生成される順序は、左右を保つ \wedge -埋め込み可能性による順序であることは容易に想像できるだろう。

3.3. WQO 理論とクラスカルの木定理

順序集合 (Q, \leq_Q) の部分集合 S が半鎖 (antichain) であるとは、任意の $p, q \in Q$ について、 $p \not\leq_Q q$ かつ $q \not\leq_Q p$ であることを意味する。この場合、しばしば $p \perp_Q q$ と書く。

定義 3.16. 擬順序集合 (Q, \leq_Q) が整擬順序 (well-quasi-order) あるいは WQO であるとは、 Q が無限下降列も無限半鎖も持たないことを意味する。

この概念は整列擬順序と訳されることが多いが、整列という用語は混乱を招く可能性があるので、本稿では用いないこととする。使わない用語は説明しないが、混乱を招く理由として、以下のような状況があるためである。



ところで、ラムゼーの定理 (*Ramsey's theorem*) と呼ばれるものがある。最も単純なケースでは、人間が6人集まれば「知り合い同士であるような3人」あるいは「互いに知り合いでない3人」のどちらかは必ず存在する、というものである。あるいは、人々が星空の中に星座という形状を見出してきた、というのもラムゼー理論の心を表す。スローガンの的には、これは

十分に巨大な構造の中には必ず秩序がある

と説明されることがある。ただ、この十分に巨大というのが本当に巨大でなければならないことが多く、実際、冒頭で述べたように、グラハム数はラムゼー理論において用いられた数である。

さて、ラムゼーの定理の最初の例のように、集団 X の各ペアを複数の属性に分割しよう。つまり、たとえば各ペアに対して「知り合いである」「知り合いでない」という属性を付けるような感じである。集合 X の大きな n の部分集合全体を $[X]^n$ と表すことにすれば、ペアへの属性付けは、関数 $c: [X]^2 \rightarrow E$ を指定することと同等である。本稿で用いるものは、「知り合いである」「知り合いでない」というような二択の設定でよいので、関数 $c: [X]^2 \rightarrow \{0, 1\}$ を考える。最初の例であれば、 X が6以上の要素を含むならば、3以上の要素を含む $Y \subseteq X$ が存在して、 Y に属すペアはいずれも同じ属性である。つまり、以下が成立している。

$$(\exists i \in \{0, 1\})(\forall x, y \in Y) c(\{x, y\}) = i. \tag{6}$$

さて、本稿で用いるものは、無限ラムゼーの定理というものである。無限に大きい構造には無限に大きな秩序がある。もし X が無限集合ならば、上記のような $Y \subseteq X$ を無限集合として取ることができる、という主張である。形式的には、任意の関数 $c: [X]^2 \rightarrow \{0, 1\}$ について、ある無限集合 $Y \subseteq X$ が存在して、式 (6) が成立する。この定理は、もともとフランク・ラムゼー (Frank P. Ramsey) が形式論理学の研究の中で証明した主張である。この無限ラムゼーの定理は認めて、次の証明を与えることにしよう。

命題 3.17. 擬順序集合 (Q, \leq_Q) が WQO であることと、次の条件を満たす Q の無限列 q_1, q_2, q_3, \dots が存在しないことは同値である。

$$(\forall m, n) [m < n \implies q_m \not\leq_Q q_n].$$

上記の式を満たす無限列 q_1, q_2, q_3, \dots は悪列 (*bad sequence*) と呼ばれる。一応確認しておくとして、この式は、クラスカルの木の定理 1.6 や関数 tree および TREE の定義で見たものと同じである！

Proof. WQO でないならば、無限下降列または無限反鎖を持つ。どちらにせよ、悪列である。

逆に, Q は悪列 $\{q_n\}_{n \in \mathbb{N}}$ を持つと仮定する. いま, $n < m$ について, 以下のようにペアに属性 $c: [\mathbb{N}]^2 \rightarrow \{0, 1\}$ を与える.

$$c(\{n, m\}) = 1 \iff q_n \geq_Q q_m.$$

このとき, 無限ラムゼーの定理より, ある無限集合 $I \subseteq \mathbb{N}$ が存在して, I に属す番号のペアは等しい属性を持つ. もし, c の I 上の値が 1 であれば, c の定義と $\{q_n\}_{n \in \mathbb{N}}$ が悪列であるという条件より,

$$(n, m \in I \text{ and } n < m) \implies (q_n \geq_Q q_m \text{ and } q_n \not\leq_Q q_m) \iff q_n >_Q q_m$$

であるから, $\{q_n\}_{n \in I}$ は無限下降列を与える. もし, c の I 上の値が 0 であれば, c の定義と $\{q_n\}_{n \in \mathbb{N}}$ が悪列であるという条件より,

$$(n, m \in I \text{ and } n < m) \implies (q_n \not\geq_Q q_m \text{ and } q_n \leq_Q q_m) \iff q_n \perp_Q q_m$$

であるから, $\{q_n\}_{n \in I}$ は無限反鎖を与える. 以上より, Q が悪列を持つならば, Q は無限下降列または無限反鎖を持つことが示された. \square

命題 3.17 より, WQO とは, 悪列を持たない擬順序である. つまり, どんな無限列 (q_i) が与えられても, ある $n < m$ が存在して, $q_n \leq_Q q_m$ となる. この性質の良いところは, \leq_Q にしか言及しないため, WQO の定義が単調であることを明示化するという点にある. 具体的には, 同じ集合 X 上の擬順序 \leq_P が擬順序 \leq_Q より粗い, つまり

$$x \leq_Q y \implies x \leq_P y$$

という条件を満たす場合を考えよう. このとき, $q_n \leq_Q q_m$ ならば必ず $q_n \leq_P q_m$ も保証される. したがって,

$$(X, \leq_Q) \text{ が WQO} \implies (X, \leq_P) \text{ も WQO}$$

という性質が成り立つのである. WQO の最初の定義 3.16 のように, 定義が無限下降列と無限反鎖に分解されていると, この性質は非自明である. この性質は命題 3.17 の恩恵のひとつである. これを利用すると, 不等式理論 Q と Q' があつたとき, もし Q' が Q より強いならば,

$$Q[A] \text{ が WQO} \implies Q'[A] \text{ も WQO}$$

ということも導ける. したがって, 可能な限り弱い不等式理論において WQO 性を示す, ということが重要である.

無限ラムゼーの定理のもうひとつの帰結を証明しよう.

補題 3.18. 擬順序 Q が WQO ならば, Q の任意の列 q_0, q_1, q_2, \dots は広義単調増大な無限部分列を含む. つまり, 増大関数 $r: \mathbb{N} \rightarrow \mathbb{N}$ が存在して, 以下が成立する.

$$i \leq j \implies q_{r(i)} \leq q_{r(j)}.$$

Proof. Q の任意の列 q_0, q_1, q_2, \dots が与えられていると仮定する．いま, $n < m$ について, 以下のようにペアに属性 $c: [\mathbb{N}]^2 \rightarrow \{0, 1\}$ を与える．

$$c(\{n, m\}) = 1 \iff q_n \leq_Q q_m.$$

このとき, 無限ラムゼーの定理より, ある無限集合 $I \subseteq \mathbb{N}$ が存在して, I に属す番号のペアは等しい属性を持つ．もし c の I 上の値が 0 であれば,

$$(n \in I \text{ and } n < m) \implies q_n \not\leq_Q q_m$$

であるが, I は無限集合なので, つまり, これは悪列であるような部分列を持つということである．しかし, Q は WQO であったから, 悪列は持たないので, これはあり得ない．したがって, 必ず c は I 上で値 1 を取るが, このとき,

$$(n \in I \text{ and } n < m) \implies q_n \leq_Q q_m$$

が成立している．無限集合 I の元を添字に持つ要素を順に並べていけば, これは $(q_n)_{n \in \mathbb{N}}$ の広義単調増大な無限部分列を与える． \square

さて, 全順序には整礎部分 (well-founded part) という概念がある．これは全順序のうち整礎になっているギリギリ限界の部分までを取ってくる操作で, 整列順序を扱う際には重要な手法である．これは, ギリギリ整礎になっていない部分を取ってくるという行為と表裏一体である．このアイデアを拡張して, 擬順序に対する WQO 部分という概念も考えたい．しかし, ここで考えるものはその裏の姿で, ギリギリ WQO になっていない部分である．この概念は WQO 理論においては, 極小悪列という名で呼ばれる．本稿では, 項上の擬順序しか考えないので, そのようなものに対する極小悪列を導入しよう．

定義 3.19. 項上の擬順序 Q 上の極小悪列 (*minimal bad sequence*) とは, 項の悪列

$$t_0, t_1, t_2, \dots, t_{i-1}, t_i, t_{i+1}, t_{i+2}, \dots$$

であり, 任意の $i \in \mathbb{N}$ および t_i の任意の真部分項 t'_i に対して,

$$t_0, t_1, t_2, \dots, t_{i-1}, t'_i, s_{i+1}, s_{i+2}, \dots$$

が悪列となるような s_{i+1}, s_{i+2}, \dots は存在しないことを意味する．

補題 3.20. 不等式理論 Q において擬順序 A から生成される擬順序 $Q[A]$ が WQO でない, つまり悪列を持つならば, 必ず極小悪列を持つ．

Proof. まず, $Q[A]$ は WQO でないので, 悪列 u_0, u_1, u_2, \dots をもつ．最初に, u_0 の部分項 u'_0 のうち, 悪列 u'_0, v_1, v_2, \dots に拡張可能であるような極小のものを取る．不等式理論の項 u_0 は有限個の記号の組合せで作られているので, u_0 の部分項は有限種類しか存在しない．したがって, 有限種類の u'_0 の悪列への拡張可能性を確認すればよいから, そのような u'_0 は存在するので, それを t_0 とおく．いま, t_0, v_1, v_2, \dots を得ているので, 次は v_1 の有限種類の部分項 v'_1 を見て, t_0, v'_1, \dots が悪列に拡張可能な極小のものを取り, それを t_1 とおく．これを繰り返せば, 極小悪列 t_0, t_1, t_2, \dots を得る． \square

WQO 理論の主結果は、既に知っている WQO をタネにして不等式理論を経由して擬順序を拡張しても、まだ WQO であることが保たれる、という形式の定理群である。つまり、様々な不等式理論 Q について、

$$A \text{ は WQO} \implies Q[A] \text{ は WQO}$$

という形式の定理が成立する。実際には、不等式理論として書き表すのが難しい擬順序構成についても、類似の結果が知られている。この種の定理の代表的な例が、ディクソンの補題、ヒグマンの補題、クラスカルの木の定理である。

まずは、語上の順序から議論しよう。先ほど見たように、弱い不等式理論で WQO の保存性を示せたならば、強い不等式理論においても WQO の保存性は成立する。したがって、可能な限り弱い不等式理論で WQO の保存性を示すことが重要である。語の理論の場合、ヒグマンの補題 (*Higman's lemma*) によれば、弱化解則さえあれば、WQO の保存性が満たされる。ここで、命題 3.8 によれば、* の弱化解則のみを持つ不等式理論 Q_w は語上の埋め込み順序に対応していたことを思い出そう。

定理 3.21 (ヒグマンの補題). 語の埋め込み順序の不等式理論 Q_w は WQO 性を保つ。つまり、

$$A \text{ は WQO} \implies Q_w[A] \text{ は WQO.}$$

Proof. A は WQO であるが、 $Q_w[A]$ は WQO でないと仮定する。このとき、補題 3.20 より、 $Q_w[A]$ は極小悪列 t_0, t_1, t_2, \dots を持つ。いま、* の結合律より、項 t_i は A 上の語と同一視できたことを思い出そう。このとき、 $a_i \in A$ を t_i の先頭の記号とする、つまり、 $t_i = a_i * t'_i$ と分解する。ここで、 t'_i として空な項も認めることにする。

このとき、 a_0, a_1, a_2, \dots は A の列であるが、 A は WQO だったので、補題 3.18 より、 $(a_n)_{n \in \mathbb{N}}$ は単調増大な無限部分列 $(a_{r(n)})_{n \in \mathbb{N}}$ を含む。このとき、

$$t_0, t_1, \dots, t_{r(0)-1}, t'_{r(0)}, t'_{r(1)}, t'_{r(2)}, \dots$$

を考えると、 $t'_{r(0)}$ は $t_{r(0)}$ の部分項であるから、極小悪列の定義より、この列は悪列ではない。したがって、ある $i < j$ について、 i 番目の要素から j 番目の要素への順序関係を成立させる。これについて、3 つのパターンがある。

1. $i < j < r(0)$ の場合: このとき $t_i \leq t_j$ であるが、これは $(t_n)_{n \in \mathbb{N}}$ が悪列でなかったことを導き、矛盾する。
2. $i < r(0) \leq j$ の場合: このとき、 $t_i \leq t'_{r(k)}$ となるが、* の弱化解則より、

$$t_i \leq t'_{r(k)} \leq a_{r(k)} * t'_{r(k)} = t_{r(k)}$$

を得る。しかし、 $i < r(k)$ かつ $t_i \leq t_{r(k)}$ ということは、 $(t_n)_{n \in \mathbb{N}}$ が悪列でなかったことを導き、矛盾する。

3. $r(0) \leq i < j$ の場合: このとき、ある $k < \ell$ について $t'_{r(k)} \leq t'_{r(\ell)}$ となっているが、いま $a_{r(k)} \leq a_{r(\ell)}$ であったから、関数適用規則より、

$$\frac{a_{r(k)} \leq a_{r(\ell)} \quad t'_{r(k)} \leq t'_{r(\ell)}}{a_{r(k)} * t'_{r(k)} \leq a_{r(\ell)} * t'_{r(\ell)}} \quad \begin{array}{l} \text{(関数適用)} \\ \text{(定義)} \end{array}$$

$$t_{r(k)} \leq t_{r(\ell)}$$

を得る．このとき， $r(k) < r(\ell)$ かつ $t_{r(k)} \leq t_{r(\ell)}$ なので， $(t_n)_{n \in \mathbb{N}}$ が悪列でなかったことを導き，矛盾する．

以上より，いずれにせよ矛盾を導くので， $Q_w[A]$ が WQO でないという仮定が誤っていた．つまり， A が WQO ならば $Q_w[A]$ も WQO である． \square

それでは，本題の木に関する順序について議論する．演算が複数あるので，木の方が語よりも厄介である．先ほど述べたように，可能な限り弱い不等式理論において WQO の保存性を示すのがよい．ここでは， $+$ と $*$ について共に弱化解則しか持たない不等式理論 R_w° を扱うことにする．以下の定理は，クラスカルの木の定理 (*Kruskal's tree theorem*) と呼ばれる．

定理 3.22 (クラスカルの木の定理). 森の左右を保つ \wedge -埋め込み順序の理論 R_w° は WQO 性を保つ．つまり，

$$A \text{ は WQO} \implies R_w^\circ[A] \text{ は WQO.}$$

Proof. A は WQO であるが $R_w^\circ[A]$ は WQO でないと仮定する．このとき，補題 3.20 より， $R_w^\circ[A]$ は極小悪列 t_0, t_1, t_2, \dots を持つ．前に説明したように，右分配律を用いて，各 t_i は次のように表すことができる．

$$t_i = a_i * (u_1^i + u_2^i + \dots + u_{\ell(i)}^i) \quad \text{or} \quad t_i = u_1^i + u_2^i + \dots + u_{\ell(i)}^i.$$

ここで， a_i は定数記号であり，各 u_k^i は連結項であり，右の場合には $\ell(i) \geq 2$ であるとする．項 t_i が連結項ならば左であり，不連結項ならば右である．いま，上の極小悪列の真部分項たち

$$U = \{u_k^i : i \in \mathbb{N}, k \leq \ell(i)\}$$

が WQO をなしていることを示そう．そうでないと仮定すると， U の悪列 v_0, v_1, v_2, \dots が存在する．このとき， $v_0 = u_k^s$ の形となっているので， v_0 は t_s の真部分項である．悪列の部分列は悪列であるので， $(v_j)_{j \in \mathbb{N}}$ の中から t_s 以降の項の真部分項となっているものだけを取り出すことによって，すべての v_j は t_s 以降の項の真部分項であると仮定できる．悪列 (t_i) の極小性の仮定より，

$$t_0, t_1, \dots, t_{s-1}, v_0, v_1, v_2, \dots \tag{7}$$

は悪列ではない．さて， $v_j = u_k^n$ であったとすると，たとえば t_n が連結項だった場合には， $+$ と $*$ の弱化解を用いると，

$$v_j = u_k^n \leq u_1^n + u_2^n + \dots + u_{\ell(n)}^n \leq a_n * (u_1^n + u_2^n + \dots + u_{\ell(n)}^n) = t_n$$

を得る． t_n が不連結項だった場合も同様である．しかし，ある $n \geq s$ について $v_j = u_k^n$ であると仮定しており， (t_i) は悪列であるから， $i < s \leq n$ について $t_i \not\leq t_n$ である．これを $v_j \leq t_n$ と組み合わせれば， $t_i \not\leq v_j$ を得る．特に，列 (7) は悪列であることが導かれるが，これは仮定に矛盾する．したがって， U が WQO であることが導ける．

つづいて， t_i のうち無限個が不連結項の場合には，悪列の部分列は悪列であるので， (t_i) は不連結項だけからなる悪列であると仮定してもよい．そうでない場合には，連結項の先頭の記号の列 (a_i) を考えると，これは A の列であるが， A は WQO だったので，補題 3.18 より， $(a_n)_{n \in \mathbb{N}}$ は単調増大な無限部分列を含む．したがって，その部分に対応する (t_n) の部分列を取ってくることによって， $i \leq j$ ならば $a_i \leq a_j$ となっていることを仮定できる．

それでは, t_0, t_1, t_2, \dots の部分項のなす次のような列を考えよう.

$$u_1^0 + u_2^0 \cdots + u_{\ell(0)}^0, \quad u_1^1 + u_2^1 + \cdots + u_{\ell(1)}^1, \quad u_1^2 + u_2^2 + \cdots + u_{\ell(2)}^2, \dots \quad (8)$$

和 $+$ を語の結合だと思えば, これは U の語の列だと思えることができる. 実際, 和 $+$ は結合律と弱化規則を満たすので, この列に関する順序は U の語上の埋め込み順序 $Q_w[U]$ と完全に一致する. U が WQO であったことから, ヒグマンの補題 (定理 3.21) より, $Q_w[U]$ も WQO である. したがって, 列 (8) は悪列ではない. つまり, ある $j < k$ が存在して,

$$u_1^j + u_2^j + \cdots + u_{\ell(j)}^j \leq u_1^k + u_2^k + \cdots + u_{\ell(k)}^k$$

を導く. t_j と t_k が不連結項であった場合には, これは $t_j \leq t_k$ を意味する. もし t_j と t_k が連結項であったならば, $a_j \leq a_k$ であったから, 弱い関数適用規則を用いて,

$$\frac{a_j \leq a_k \quad u_1^j + \cdots + u_{\ell(j)}^j \leq u_1^k + \cdots + u_{\ell(k)}^k}{a_j * (u_1^j + \cdots + u_{\ell(j)}^j) \leq a_k * (u_1^k + \cdots + u_{\ell(k)}^k)} \quad (\text{弱い関数適用})$$

$$\frac{a_j * (u_1^j + \cdots + u_{\ell(j)}^j) \leq a_k * (u_1^k + \cdots + u_{\ell(k)}^k)}{t_j \leq t_k} \quad (\text{定義})$$

を得る. したがって, いずれにせよ $j < k$ かつ $t_j \leq t_k$ を得るが, これは (t_n) が悪列であるという仮定に矛盾する. よって, $R_w^\circ[A]$ は WQO でないという仮定が誤っていたということである. つまり, $R_w^\circ[A]$ は WQO である. \square

上の定理 3.22 は, 定理 1.6 で述べたクラスカルの木の定理よりも強い主張になっている. なぜなら, 命題 3.15 より, ラベル付き森の \wedge -埋め込み順序は不等式理論 R_w^- と対応していたが, 理論 R_w^- は理論 R_w° よりも強い. したがって, 定理 3.22 は R_w^- についても成立する. ここで, A を n 個の元を持つ離散順序 (つまり, サイズ n の反鎖) としよう. すると, $R_w^-[A]$ がちょうど n -ラベル付き森の \wedge -埋め込み順序に対応する. A は有限順序なので, 明らかに WQO である. よって, 定理 3.22 より $R_w^-[A]$ もまた WQO であると示される.

$$A \text{ は有限} \implies A \text{ は WQO} \implies R_w^-[A] \text{ は WQO.}$$

つまり, n -ラベル付き森の \wedge -埋め込み順序は WQO であるから, クラスカルの木 of 定理 1.6 が導かれた.

§ 4. あとがき

4.1. 歴史的リマーク

WQO 理論の歴史: WQO 理論 [8] は様々な分野で同時に発生した. WQO の定義は「有限基底」の形で定式化することもでき, 代数学の文脈ではそちらの同値な定義が用いられることも多い. WQO 理論の最も基本的な結果は, WQO が積で閉じている (積順序が WQO 性を保つ) という結果であり, これはディクソンの補題 (Dickson's lemma) と呼ばれる. ディクソンの補題は, たとえばグレブナー基底の存在性証明などに用いられることでよく知られている.

WQO 理論の次の重要なステップは、アメリカ数学協会 (MAA) の発行する雑誌 American Mathematical Monthly において 1949 年にエルデシュ (Paul Erdős) が提出した問題から発生した。この問題は、自然数列の整除順序に関する WQO 性のある種の保存性を問うものであった。1952 年、このエルデシュの問題は、何人かの数学者によって独立に解決された。そのうちのエルデシュとラド (Richard Rado) の結果 [2] は、有限集合の支配順序に関する WQO 性の保存を導くものである。この同年にヒグマン (Graham Higman) が証明したものが、ヒグマンの補題 (定理 3.21) である。ヒグマンの補題は、語の埋め込みに関する WQO 性の保存を導くものであり、本稿で説明したように、これは上述のエルデシュ-ラドの定理より強力である。ヒグマンもまた、この補題を抽象代数の文脈で提示しており、WQO 性のことを有限基底性 (*finite basis property*) と呼んでいる [6]。

純粋なグラフ理論の文脈では、1940 年の Andrew Vazsonyi が予想したとされる、有限木の同相埋め込み (homeomorphic embedding) による順序の WQO 性である。エルデシュによって、この問題はクラスカル (Joseph B. Kruskal) に伝えられ、1954 年に解決が与えられた [8]。これがクラスカルの木の定理である。用語は異なるが、これは本稿で扱った \wedge -埋め込みと同値な概念である。

その後、木を越えてより一般のグラフに興味が移ることとなる。1930 年に証明されていたクラトフスキ (Kazimierz Kuratowski) の定理もまた、グラフ理論における有限基底性の代表的な例と言ってよいかもしれない。これは、グラフが平面グラフであることと、 K_5 (5 頂点完全グラフ) や $K_{3,3}$ (6 頂点完全二部グラフ) の細分であるような部分グラフを含まないことが同値であるというものである。その後、グラフ理論においてマイナーと呼ばれる概念が導入され、平面グラフは同様の禁止マイナーによる特徴付けを得られる。この平面グラフに関する定理と、クラスカルの木の定理の共通の究極的拡張が、ロバートソン (Neil Robertson) とシーモア (Paul D. Seymour) のグラフ・マイナー定理 (*graph minor theorem*) である。これは、

有限グラフはマイナー関係の下で WQO をなす

という定理であり、グラフ理論における最も深遠な定理の一つとして知られる。1983 年から 2004 年に掛けて、総ページ数 500 を越える 20 編にわたる一連の論文 (1983 年の論文タイトルは “Graph Minor I” であり、2004 年の最後の論文のタイトルは “Graph Minor XX” であるように、通し番号で名が付けられている) を経て証明された [12] という特異性からも有名である。

独立命題と組合せ論: これらの組合せ論の定理は、ある数学基礎論の研究から、巨大数と交わることとなる。ゲーデルの不完全性定理 (Gödel’s incompleteness theorem) により、十分に算術を含む無矛盾な再帰的公理化可能理論において、肯定も否定も証明できない命題 (独立命題) が存在する。しかし、不完全性定理は、あくまで独立命題が存在するとただ漠然と主張するだけである。したがって、数学者による日常的な数学研究の過程で独立命題に遭遇することがあるかどうかは明白ではない。ところが、その後の歴史において、集合論においては連続体仮説の独立性証明を皮切りに、測度論におけるポレル予想、アーベル群論においてはホワイトヘッドの問題、近年では、作用素環論におけるカルキン環の外部自己同型の存在など、数多くの数学的問題が、ZFC と呼ばれる標準的な集合論の体系と独立であると示されてきた。

とはいえ、これらは抽象数学の命題なので、もう少し地に足の着いた算術的命題ではどうだろうか。一階ペアノ算術から独立であるような自然な算術的命題は、1977年にパリス (Jeff Paris) とハーリントン (Leo Harrington) によって与えられた。彼らが考えたものは、有限ラムゼーの定理のある種の強化であり、自然数論の言葉で記述することができるのに、自然数論の形式化であるペアノ算術では証明も反証も不可能な独立命題である。このパリス-ハーリントンの定理の独立性は、この有限ラムゼー的原理がある種の急増加関数を生み出すことに起因する。その後、1981年に、カービー (Laurie Kirby) とパリスは、グッドスタイン列という数列が0に到達するまでの長さがとてつもない急増加関数となることを発見する。これを利用して、グッドスタイン列の0への収束性もまた、ペアノ算術では証明も反証も不可能な独立命題であることが導かれた。

証明論においては、理論に対するある種の不変量として、証明論的順序数 (proof theoretic ordinal) や可証再帰関数 (provably recursive function) と呼ばれる概念が取り扱われている。このように理論内における順序数や急増加関数を分析して、理論の限界を測るという行為がなされている。一階ペアノ算術の証明論的順序数は ε_0 であることが知られている。そして、1980年代のフリードマンの発見は、クラスカルの木の本の定理が Γ_0 を越える順序数を生み出すということであった [15, 11]。 Γ_0 は一階ペアノ算術を遥かに超える算術的超限再帰の体系 ATR_0 の証明論的順序数である。クラスカルの本の定理という、グラフ理論の日常的研究の中で生まれた定理が、(パリス-ハーリントンのラムゼー的原理やグッドスタイン列の収束性を遥かに越えて) ATR_0 から証明できないという事実は、多くの数理論理学者に衝撃を与えた。

様々な算術的独立命題が巨大関数を生むということはよく知られていたが、フリードマンはより厳密な有限の独立性にも興味を抱き、それらの独立命題に具体的な有限の値を入力した値の評価にも手を付け始めたようである [4]。そのうちのひとつが、クラスカルの本の定理から作られた $TREE(3)$ である。そして、グラフ・マイナー定理はクラスカルの本の定理を包含する公理なのであるから、グラフ・マイナー定理の力はそれ以上である [3]。それを利用して、フリードマンによって生み出されたものが、サブキュービックグラフ数 (*subcubic graph number*) と呼ばれる巨大数であり、 $TREE$ よりも巨大な数を生み出すことができる。

BQO 理論: さて、もう少し語りたことがあるので、再び組合せ論に戻ろう。WQO 理論のもうひとつの方向性として、不等式理論の有限項演算を無限項演算に拡張するというものがある。つまり、不等式理論ではたとえば二項和 $+$ や二項積 $*$ を扱えたが、無限和 \sum や無限積 \prod も数学では日常的に用いられる。等式理論の無限項演算への拡張はあまり難しくなく、実際、そのような理論も深く研究されている。しかし、不等式理論における重要な問題は WQO 性の保存であり、無限項演算を持つ不等式理論について、これは一般には成り立たない。言い換えれば、WQO の閉包性はあまり良いものではない。

これを解決するものが、1965年にナッシュ-ウィリアムズ (Crispin Nash-Williams) が導入した BQO 理論 [10] である。BQO とは、ウェル (well) より良い、ベター (better) を表す。ナッシュ-ウィリアムズは、WQO の極めて技巧的な強化である BQO を導入し、本稿で扱ってきた語の理論や森の理論を無限項演算に拡張し、ヒグマンの補題やクラスカルの本の定理の無限版の証明を与

えたのである。また、BQO 理論の構築の過程において、ナッシュ-ウィリアムズは開閉ラムゼー (clopen Ramsey) と呼ばれる性質を証明し、これがその後のポレル・ラムゼー定理や、集合論におけるラムゼー性公理の研究を誘発したという点でも重要である。

とはいえ、語の理論や森の理論を無限項演算に拡張する動機が分からないという人も多いただろう。ナッシュ-ウィリアムズの BQO 理論が当時注目を浴びた最初の切っ掛けは、1971 年のレイヴァー (Richard Laver) によるフレーゼ予想 (*Fraïssé conjecture*) の解決 [9] であった。1948 年、フレーゼ (Roland Fraïssé) は次の予想を打ち立てた。

可算全順序たちは埋め込み関係の下で WQO をなす。

フレーゼは WQO という言葉は使っていないが、とにかく WQO 理論の出番のようである。しかし、この問題に対して、WQO 理論は無効であった。その理由を軽く説明しよう。このためにハウスドルフ (Felix Hausdorff) による可算全順序の解析を利用する。これは可算全順序の整列和と逆整列和を考えるものである。

$$\begin{aligned} L_0 + L_1 + L_2 + \cdots + L_\alpha + L_{\alpha+1} + \cdots \\ \cdots + L_{\alpha+1} + L_\alpha + \cdots + L_2 + L_1 + L_0 \end{aligned}$$

これは、可算整列和 \sum と可算逆整列和 \sqsupset という 2 種類の無限項演算を持つ体系 L を考えることになる。まず、1 点全順序を起点として、理論 L によって生成される全順序 L_1 (つまり、1 点全順序に整列和と逆整列和を組み合わせて作られる項全体上の順序) を考える。次に L_1 から L -生成される全順序 L_2 を考える。このように、整列和と逆整列和を用いて作った項を入れ子にしてどんどん積み上げていく。そして、最小の非可算順序数 $\Omega = \omega_1$ について、 $(L_\alpha)_{\alpha < \Omega}$ を考えよう。有理数の順序型を含まない可算全順序は、必ずこのようにして作り上げていくことができる。したがって、可算全順序の分析に必要なものは無限項演算を分析する道具であり、それが BQO 理論であった。そして、レイヴァーは BQO 理論を巧みに利用し、このフレーゼ予想を見事に肯定的に解決したのである。

他にも無限項演算の使い所は多い。たとえば、急増加階層などの順序数単体ではなく梯系 (基本列) に依存する概念においては、梯子こそが順序数の基本データと思ふべきである。そして、たとえば無限上昇列上の支配順序の理論 O を考えよう。1 点集合を起点として、 $(O_\beta)_{\beta < \alpha}$ から O -生成される擬順序を O_α とすれば、これが順序型 α 以下の順序数の梯系を直接与えるシステムとなっている。

至る所に木の順序: 最後に、木の順序構造に関する筆者の経験について語っておきたい。思い返せば、クラスカルの木定理と数理論理学にまつわる話はかなり昔から知っていた—が、しかし、クラスカルの木定理が一体何の役に立つ定理なのかさっぱり知らなかった。そういうわけであるから、木の順序とはまあ面白い茶飲み話であるが、自分にはあまり関わりのない話であろう。そう長らく思っていたのである。

ところが、最近、身の回りでも、木の順序によく遭遇する。というのも、ある種の関数には、しばしばその関数の振る舞いを制御する木構造が背後に隠されているようである。たとえば、関数を計

算するアルゴリズムを記述するフローチャートのようなものを思い浮かべてもらうとよい．このように背後に木状の制御機構を持つような関数の比較など（ある関数で別の関数をシミュレートできるか、など）を考える際に、「木の順序」が突如として出現してくるのである．そのような典型的な例として、たとえば計算理論におけるヘルトリング (Peter Hertling) の研究がある．彼は、BSS 機械という実数上の計算モデルの計算能力の分析の過程で、その計算構造の中に有限森の準同型順序が隠れ潜んでいることを発見したのである．

筆者が最近、大きな興味を持っているものは、記述集合論においてふと現れた、木の順序構造である．その第一ステップとして、セリヴァノフ (Victor Selivanov) は、「ベール 1 級分割の位相的分類の順序構造」がある種の森の準同型順序と同型であることを示している [13, 14]．この結果を拡張して、モンタルバン (Antonio Montalbán) と筆者 [7] は、「ボレル可測関数の位相的分類の順序構造」が、森の不等式理論 R (の和 $+$ を無限項演算 \sum に拡張したもの) にヴェブレン関数 ϕ に関する公理を加えた理論における準同型順序（つまり、弱化和縮約を付加して証明できる擬順序関係）と同型であることを示した．ここで、ラベル付き森の理論におけるヴェブレン関数の直感的意味は、以下のようなものである．まず、 ϕ_0 は、森をラベルに変える演算である．これを利用すると、

森でラベル付けされた森でラベル付けされた森でラベル付けされた森...

のような入れ子構造を得られる．無限和 \sum を利用すると、このラベルの入れ子を超限的に続けることができる．つづく ϕ_1 は、この森のラベルの超限入れ子の不動点を並べる演算である．ただし、順序数に対するヴェブレン関数と違って、 $\phi_1(t)$ の t はこのヴェブレン的理論における項である．つまり、 t は森でラベル付けされた森でラベル付けされた森のようなものだったりするので、“ t 番目の不動点” の概念は、順序数のヴェブレン関数ほど単純な概念ではない．したがって、 ϕ_1 の不動点を取る演算 ϕ_2 の意味を説明するのは難しいが、とにかく、通常のヴェブレン階層のように

$$\phi_0, \phi_1, \phi_2, \phi_3, \dots, \phi_\omega, \phi_{\omega+1}, \dots$$

が作られていく．そして、 $\Omega = \omega_1$ を最小の非可算順序数として、 $(\phi_\alpha)_{\alpha < \Omega}$ まで考える．すると、このようにして作られたヴェブレン的構造が、ボレル可測関数の位相複雑性の構造と ぴったり一致 する（より正確には、そのままでは互いに擬順序なので商を取って半順序にする必要があるが、そうしたときに、このヴェブレン的構造とボレル可測関数の位相的分類の順序構造が順序同型であることが示せる）．ただのラベル付き木ですら非常に複雑なのに、それにさらにメタ的にヴェブレン階層を重ねていった、こんなにも複雑な構造が、ボレル可測関数の位相的分類という極めて自然な数学的活動の中から出てくる、というのが非常に面白いところである．

4.2. 参考書籍

▶ 本稿では TREE 周辺の話しか取り扱っていないが、巨大数論の世界は広大である．とりあえず巨大数論を一望したいのであれば、以下の書籍が最もよくまとまっていると思われる．

フィッシュ, 巨大数論 第2版, NextPublishing Authors Press, 2017.

計算可能レベルの話は非常によくまとまっているが, ただし計算不可能レベルの話は少し怪しいので, 計算不可能レベルの部分はあまり真に受けないでほしい. ただ, それでも計算可能関数までのパートは非常によく書かれているので, ぜひ読んでほしい書籍である.

▶ もっと気楽に巨大数に入門したい場合には, 巨大数をテーマにした漫画も出版されている.

小林銅蟲, 寿司 虚空編, 三オブックス, 2017.

漫画であるものの, わりと巨大数について正確に書かれている. 小林銅蟲氏は他にも様々な漫画を書いており, 2020年4月現在は, 数学セミナーという雑誌で「せいすうたん」という整数漫画を連載中である. 一般誌では, 現在は飯漫画を連載しているらしい(未読). ちなみに私は大昔に「ねぎ姉さん」というウェブ漫画で小林銅蟲氏のことを知った.

▶ 古代から前世紀始め頃までの巨大数にまつわる歴史がまとまっている文献としては以下がある.

鈴木 真治, 巨大数 (岩波科学ライブラリー), 岩波書店, 2016.

話題の範囲が幅広く, 非常に勉強になる. しかし, 時代が現代に近づいてきて, 数理論理学が絡んでくる内容になってくると, 特に巨大数小史 pdf の方にはかなり賛同できない文章が散見されていたので紹介するかどうか迷ったが, 書籍版ではその多くが削除されていたので書籍版を紹介しておく.

▶ 日本評論社の発行する「数学セミナー」という雑誌では, 2019年7月号において巨大数が特集された.

数学セミナー 2019年7月号 特集「おおきな数」, 日本評論社, 2019.

私は「アッカーマン関数とヒルベルト」というテーマの記事を寄稿している. その他にフィッシュ氏による巨大数論の記事の他, 岡本吉央氏によるグラハム数の記事や酒井拓史氏による集合論の記事なども収録されている.

▶ 青土社の発行する「現代思想」という雑誌では, 2019年12月号において巨大数が特集された.

現代思想 2019年12月号 特集=巨大数の世界, 青土社, 2019.

私は「無限の名を呼ぶ 巨大関数を取りまく数学小史」というテーマの記事を寄稿している. その他にフィッシュ氏による巨大数論の記事の他, 藤田博司氏や淵野昌氏による集合論の記事, 徳重典英氏による組合せ論(ラムゼー理論, セメレディの定理など)の記事なども収録されている. 小林銅蟲氏や詩野うら氏による書き下ろし漫画も掲載されている.

▶ WQO 理論に関して和書は存在しないと思われる. WQO 理論の入門的文献としては, 以下の pdf が丁寧に書かれている.

照井 一成, NASH 村とスライム退治: 整列擬順序入門,
<http://www.kurims.kyoto-u.ac.jp/~terui/zengaku2018.pdf>

同著者による以下の一般向け書籍にも, 本稿と関わりの深い内容が書かれている.

照井一成, コンピュータは数学者になれるのか? -数学基礎論から証明とプログラムの理論へ-, 青土社 2015.

一般向けの啓蒙書であるが, 数学基礎論の基本的話題が網羅されており, 素人でも専門家でも誰でも一見の価値はあると思う. 私がこれまでの人生で見た中で最高の数学基礎論の啓蒙書であり, 手放してオススメできる. ちなみにタイトルだけを見ると, 数学基礎論に関してそんなに広範な話題が書かれているとは到底予想できなかったの, つい最近まで完全にスルーしていた.

▶ 数学基礎論のその先の段階の和書としては, 以下を紹介しておく.

田中 一之, 数学基礎論序説: 数の体系への論理的アプローチ, 裳華房 2019.
新井 敏康, 数学基礎論, 岩波書店 2011.

問題 10. Numberphile の以下の動画を視聴して, 感想を書け(英語の勉強にもなるかもしれない).

- The Enormous TREE(3) <https://www.youtube.com/watch?v=3P6DWAwwViU>
- TREE(3) (extra footage) <https://www.youtube.com/watch?v=IihcNa9YAPk>

参考文献

- [1] Hyp cos. tree function and tree(3). [https://googology.wikia.org/wiki/User_blog:Hyp_cos/tree_function_and_TREE\(3\)](https://googology.wikia.org/wiki/User_blog:Hyp_cos/tree_function_and_TREE(3)).
- [2] Paul Erdos and Richard Rado. Advanced Problems and Solutions: Solutions: 4358. *Amer. Math. Monthly*, 59(4):255–257, 1952.
- [3] Harvey Friedman, Neil Robertson, and Paul Seymour. The metamathematics of the graph minor theorem. In *Logic and combinatorics (Arcata, Calif., 1985)*, volume 65 of *Contemp. Math.*, pages 229–261. Amer. Math. Soc., Providence, RI, 1987.
- [4] Harvey M. Friedman. Long finite sequences. *J. Combin. Theory Ser. A*, 95(1):102–144, 2001.
- [5] Jean H. Gallier. What’s so special about Kruskal’s theorem and the ordinal Γ_0 ? A survey of some results in proof theory. *Ann. Pure Appl. Logic*, 53(3):199–260, 1991.
- [6] Graham Higman. Ordering by divisibility in abstract algebras. *Proc. London Math. Soc.* (3), 2:326–336, 1952.
- [7] Takayuki Kihara and Antonio Montalbán. On the structure of the Wadge degrees of bco-valued Borel functions. *Trans. Amer. Math. Soc.*, 371(11):7885–7923, 2019.

- [8] Joseph B. Kruskal. The theory of well-quasi-ordering: A frequently discovered concept. *J. Combinatorial Theory Ser. A*, 13:297–305, 1972.
- [9] Richard Laver. On Fraïssé’s order type conjecture. *Ann. of Math. (2)*, 93:89–111, 1971.
- [10] C. St. J. A. Nash-Williams. On well-quasi-ordering infinite trees. *Proc. Cambridge Philos. Soc.*, 61:697–720, 1965.
- [11] Michael Rathjen and Andreas Weiermann. Proof-theoretic investigations on Kruskal’s theorem. *Ann. Pure Appl. Logic*, 60(1):49–88, 1993.
- [12] Neil Robertson and P. D. Seymour. Graph minors. XX. Wagner’s conjecture. *J. Combin. Theory Ser. B*, 92(2):325–357, 2004.
- [13] Victor L. Selivanov. Hierarchies of Δ_2^0 -measurable k -partitions. *MLQ Math. Log. Q.*, 53(4-5):446–461, 2007.
- [14] Victor L. Selivanov. Extending Wadge theory to k -partitions. In *Unveiling dynamics and complexity*, volume 10307 of *Lecture Notes in Comput. Sci.*, pages 387–399. Springer, Cham, 2017.
- [15] Stephen G. Simpson. Nonprovability of certain combinatorial properties of finite trees. In *Harvey Friedman’s research on the foundations of mathematics*, volume 117 of *Stud. Logic Found. Math.*, pages 87–117. North-Holland, Amsterdam, 1985.