ワークショップ資料

野矢茂樹

以下はあくまでも資料ということで、『論理哲学論考を読む』（野矢茂樹、哲学書房、２００２年）からの抜粋である。当日はここに含まれている論点を整理して発表する。

資料１

4–4　論理形式の解明はどのようにするか

前章で論じた名の論理形式の解明について、さらに詳しく見ていかねばならない。

『論考』においては、前章までに見てきたように、事実は対象の配列であり、命題は名の配列にほかならない。そして名は対象の代わりとなる。ここにおいて、すべては一律に「名」として捉えられている。思わせぶりな言い方をして申し訳ないが、なぜウィトゲンシュタインは、固有名や述語といった区別を表立たせることを避け、すべてをただ「名」として捉えようとしたのか、その真の意義を理解するには、われわれはまだ多くのことを論じなければならない。しばし、待たれよ。

ともあれ、名はさらに固有名や性質語、関係語に区別される。また、そうした品詞カテゴリーとは別に、ある名には、意味上それに特有の結合可能な名の範囲がある。そこでたとえば、「ポチは白い」は有意味な命題であるが、「白いは重い」とか「ウィトゲンシュタインは２で割り切れる」といった命題は無意味とされる。では、こうした論理形式はどのようにして明確化されるのだろうか。それがここで見ておかねばならない問題であり、後でタイプ理論批判とパラドクスの解決につながる部分にほかならない。

ここで、議論を見やすくするために、きわめて単純な言語を考えよう。名は、「ポチ」「ミケ」「富士山」「白い」「走る」「噴火する」だけとする。助詞は省略し、「ポチ - 白い」のように名を配列させてポチは白いという事態を表現する。

解明は、この言語において有意味な命題を使用してみせることによって為される。ただし、われわれが日常言語においてそうしているように何気なくただ使っているだけでは解明にはならない。何気なく使いこなすことは十分にできるということを背景として、とくにその論理形式に照明を当てるよう工夫しなければならない。そこでウィトゲンシュタインが行なう方法は、有意味な命題において、論理形式を問題にしたい名を残し、他を空欄にするというやり方である。

「ポチ」を例にとろう。「ポチ - 白い」という命題は有意味であるから、ポチを残してこのように空欄つきの表現を作る。

ポチ - x

これは関数表現である（「ポチ関数」と呼ぼう）。変項 x には名が入る。この x に入力可能な名は「白い」と「走る」である。これが、ポチ関数の定義域となる。そして、それらを入力したときに出力されるのは、「ポチ - 白い」「ポチ - 走る」という命題である。これが、ポチ関数の値域となる。

ポチ関数「ポチ - x」－定義域、{「白い」、「走る」}
値域、{「ポチ - 白い」、「ポチ - 走る」}

同様に、ミケ関数「ミケ - x」も同じ定義域をもち、ポチの場合と同型の命題を値域とする。このことは、名「ポチ」と名「ミケ」が同じ論理形式をもっていることを示している。

他方、富士山関数「富士山 - x」の場合には、「富士山 - 白い」はよいが、「富士山 - 走る」は無意味となる。そこで、定義域は {「白い」、「噴火する」} であり、値域は {「富士山 - 白い」、「富士山 - 噴火する」} である。

このことは、名「富士山」の論理形式が「ポチ」や「ミケ」とは異なることを示しているが、同時に、「ポチ」も「ミケ」も「富士山」も「a - x」という形の関数であり、品詞としては固有名であることを示している。

もうひとつ、「白い」も見ておこう。この場合には、「ポチ - 白い」のような有意味な命題において「白い」を残して他を空欄にし、「x - 白い」とする。この関数の定義域は {「ポチ」「ミケ」「富士山」} である。そして、関数が「x - a」という形をしていることは、「白い」という名が性質語であることを示している。

いささか単純化されすぎてはいるが、基本的に、これが解明の具体的手順であると考えられる。

ここで、これがフレーゲ的な関数とはまったく異なるという点を押さえておかねばならない。フレーゲの関数は、たとえば「x は白い」の場合、そこに入力されるものは「ポチ」という名が指示する対象であり、出力されるものは真偽であった。つまり、フレーゲにおいて、関数が働く場所は言語の外、そこにいるポチやミケという対象なのである。あるいは、それはラッセル的な命題関数とも異なる。ラッセルの場合には、入力はフレーゲと同様に対象であるが、出力は真偽ではなく、命題である。しかも、ラッセルの言う「命題」とはウィトゲンシュタインの「命題」があくまでも言語表現であったのに対し、むしろその言語表現の意味である。それゆえラッセルの場合にも、命題関数とは言語表現そのものにおいてではなく、その意味する対象や命題内容において働くものとなっている。

他方ウィトゲンシュタインの場合はそうではない。ウィトゲンシュタインの関数は論理形式の解明のために工夫された手段であり、あくまでも言語の中でのみ働く。定義域はポチやミケという対象ではなく、「ポチ」や「ミケ」、つまり名である。そして値域は真偽ではなく、「ポチ - 白い」といった命題とされる。それはラッセル的な言語表現の意味内容ではなく、「ポチ - 白い」という言語表現そのものにほかならない。

別の言い方をするならば、フレーゲにとって関数とは世界の対象とともに働く実質をもった道具立てであり、だからこそ、関数それ自身も対象として再び関数の入力項になりえたのである。しかし、『論考』の場合は関数はただ言語のあり方を整理するための便法にすぎない。それはそれ自身対象となりうるような実質をもたない。ただの入力項たる名と出力項たる命題の対照表にすぎない。いわば、ウィトゲンシュタインは関数を徹底的にノミナルに捉えるのである。

私の解釈が正しいならば、ウィトゲンシュタインは固有名さえも、「ａ - ｘ」のように関数化してその論理形式を解明しようとするのだが、これなどもまったく命題関数の考え方にはそぐわないものである。フレーゲによれば、関数表現は空欄をもち、それゆえ補完される必要があるが、固有名は関数表現を補完するものではあっても、それ自身は他の何かによって補完される必要のない飽和した表現にほかならない。それに対してウィトゲンシュタインの場合には、述語が固有名によって補完されねばならないように、固有名もまた述語によって補完されねばならない。固有名と述語とは、相補的な、ともに不飽和な表現なのである。

さあ、これで準備は整った。ラッセルのパラドクスに向かうことにしよう。

4–5　パラドクスの解決

タイプ理論が主張するようなタイプの階層性が名に認められるのかどうか、この点について『論考』の考えははっきりしない。しかし、かりにタイプの階層性が名に存することが認められるにせよ、ラッセルがそのために示した道はまったく転倒しているというのが、ウィトゲンシュタインの最初の論点である。

ラッセルの階層性のアイデアは、まず個体をタイプ０とするところから始まっていた。そしてそれに依拠して、個体に対する述語をタイプ１とし、以下同様に進むのである。するとわれわれは述語のタイプ分けを捉えるのに、まず個体という対象のレベルを把握しなければならない。だが、これは前章で示した『論考』の道の正反対である。前章の議論を思い出していただきたい。対象に到達するのにわれわれは言語全体に熟達しなければならなかったのである。対象は分析の結果であり、出発点ではありえない。素朴には、われわれは対象を、とくに個体を把握するところからすべてを出発させるよう

にも思われる。だが、『論考』冒頭の主張がここに響き渡る。「世界は事実の総体であり、物の総体ではない。」われわれは物から出発することはできないのである。

> 三・三三一　ラッセルの誤りは、記号の規則を立てるのに記号の指示対象を論じなければならなかった点に示されている。

ラッセルはウィトゲンシュタインのこの批判を理解できなかったのではないだろうか。同じ土俵の上で技をかけあっているというよりは、根本的に土俵が違っているという感じである。

名にタイプの階層性があるとして、それはつまり名の論理形式に属することにほかならない。それゆえそれは名の解明によって明らかにされねばならない。具体的には、前節で述べたような関数による分析を遂行することによる。ここで、先にウィトゲンシュタインが提案していた関数（ポチ関数等々）が、命題関数とは決定的に異なる点を思い出してほしい。命題関数の入力項は個体である。しかし、ウィトゲンシュタインが解明に用いる関数はそのような言語外の対象をいっさい要請しない。あくまでも言語の中にとどまり、われわれの言語使用のあり方を整理する道具立てにほかならない。すなわち、「ポチ－ｘ」等の穴あき命題は、名を定義域とし命題を値域とする、言語内的な関数なのである。かくして、名にタイプの階層性が認められるとしても、それは個体などというものに訴えるのではなく、われわれの言語使用のあり方を分析することによるのでなければならない。

名にタイプがあることが認められるならば、それは次のように解明されうるだろう。ただし、議論を簡単にするために、関係語は考慮からはずすことにする。

まず、名「ａ」がタイプ０、すなわち固有名であることは、「ａ－ｘ」という関数のみがその論理形式を特徴づけることに示される。たとえば「ポチ」は、「ポチ－ｘ」という関数として、その定義域と値域で論理形式が示される。そのとき、定義域には「白い」や「走る」のようなタイプ１の名、すなわち性質語が含まれ、タイプ０の名は含まれることはない。つまり、「ポチ－走る」は有意味であるが、「ポチ－ミケ」は無意味である。

次に、名がタイプ１であることは、定義域がタイプ０の名のみで成り立っていることに示される。タイプ１の名はもっともタイプの低い性質語であり、それゆえ固有名（タイプ０）を主語にとることになる。

以下同様に、名がタイプ２であることは、定義域がタイプ１より低い名だけで成り立っていることに示される。一般に、タイプ１以上の名に関して、名がタイプｎであることは、定義域がタイプｎより低いタイプの名のみで成り立っていることに示されている。

もしすべての名がタイプをもつのであれば、いまの規定から、すべての名はタイプの階層性を自動的にもつことになる。しかし、いま述べたことは名がタイプｎであることの定義であるから、まだすべての名にタイプが与えら

れるかどうかは分からない。たとえば名ａの定義域に名ｂが登場しているとしよう。そのとき、もし名ａがタイプｎであるならば、名ｂはそれより低いタイプｍでなければならない。ところが、ここで名ｂの定義域に名ａが登場するというようなことが起こっていたとする。そのとき、こんどは逆に、名ａは名ｂよりも低いタイプでなければならなくなる。もしそういう入れ子状況が起こっているならば、そのときは、名ａにも名ｂにもタイプを与えることができない。つまり、ａをｂで述語づけることも、逆にｂをａで述語づけることもできるとしたならば、その場合にはａとｂの間にタイプの階層性を与えることはできず、それゆえａとｂはタイプをもたない名ということになる。

あるいは、もっとダイレクトには、名ａの定義域に名ａ自身が含まれている場合である。つまり、ａをａ自身で述語づけるという自己言及が発生している場合であるが、この場合にも、名ａにはタイプを与えることができない。しかし、それはそれ自体としてはとくに困ったことというわけでもないだろう。たとえば、「曖昧である」という名を考えるならば、その定義域に自分自身が含まれているというのは、おおいにありそうなことである。自己言及文のすべてが悪さをするわけではない。無害な自己言及文もある。

とすれば、ラッセルのパラドクスはどうなってしまうのだろう。

だって、よろしいか。ある名の定義域に自分自身が含まれている場合に自己言及が生じるわけだが、定義域に含まれているということは、その結果として出力される自己言及文が有意味な命題として値域に含まれていることを意味している。つまり、その自己言及文はなんらパラドクスを引き起こすものではないはずである。逆に、その自己言及文が無意味であるならば、それは値域に含まれていないということであり、それゆえ定義域にも自分自身は含まれていないのである。とすれば、その場合にはパラドクスを引き起こす文を作ることはできないことになる。いずれの場合でも、パラドクスの生じる余地はもはや残されていない。

なんだか、「あれ？」という感じである。ウィトゲンシュタインの観点からすれば、パラドクスは自然消滅しているのである。どこに結び目があったのだろうといぶかしく感じられるほど、パラドクスがはらりとほどけている。

もう一度、くりかえそう。自分自身が定義域に含まれているときには、その結果生じる自己言及文が有意味であることは保証されている。もしその自己言及文が無意味ならば、定義域から自分自身を排除すればよい。解明とはそういうことである。しかし、定義域に自分自身が含まれていないならば、自己言及文を作ることはできない。かくして、ラッセルのパラドクスは生じない。

パラドクスの発生地点と根本的に異なるところに立っているため、その根っこの違いを認めてしまえば、あとはあっけないのである。では、その根っこの違いとは何だったのか。このあっけなさの仕掛けを捉えよう。

ウィトゲンシュタインは関数を完全にノミナルに捉えようとする。それは

入出力の対照表にすぎない。定義域が異なれば、それはもはや同じ関数とはみなされないのである。このことの気分をつかむために、命題関数ではないが、「xの母親」という関数を考えてみよう。そしてその定義域は人間とする。そこでたとえばxにルートヴィヒを入力すると、その母親であるレオポルディーネが出力される。他方、「xの母親」という表現は同じであるが、その定義域を霊長類とする関数を考えよう。その場合もxにルートヴィヒを入力するとレオポルディーネが出力されるが、こんどはxにチンパンジーのアユムを入力すると母親のアイが出力される。さて、両者の場合に「xの母親」は同じ関数なのだろうか。ひとつの答えは「同じ関数が異なる定義域をもつ」というものである。そしてもうひとつの答えは、ウィトゲンシュタインの答えであり、「両者は関数として異なっている」というものである。

命題関数「xは神経質である」で考えてみよう。日常言語においてわれわれは、暗黙の内に、その定義域を人間ないしはある程度人間と生活の仕方が似ていると感じられる動物に限定しているだろう。そこでそれが暗黙の内であることに乗じて、あるトマト（トマちゃん）について、「トマちゃんは神経質だ」と言ったとする。ウィトゲンシュタインならば、それは無意味だと言うだろう。他方、命題関数が定義域と独立に定まっていると考えるならば（あるいはすべての個体に関して定義されていると考えるならば)、「トマちゃんは神経質だ」は無意味ではなく、偽であるということになる。そのとき、「トマちゃんは神経質ではない」は真となる。それに対して、ウィトゲンシュタインならば、「トマちゃんは神経質ではない」もまた無意味となる。

どちらにも分があるように思われる。人間であろうとチンパンジーであろうと「母親」という概念は同じであるように思われるため、関数としても同一であるような気もする。「トマちゃんは神経質だ」は無意味な気もするが、「トマちゃんは神経質ではない」は真であると言いたくもなる。さらに言えば、そんなことはどっちでも大差ないように思われもするのではないだろうか。しかし、大違いなのである。

関数を定義域と独立に捉えるという考え方は、自己言及を無制限に考えてしまうことにつながる。すなわち、なんであれ、定義域Ｄをもつ関数Ｆにおいて、その関数それ自体をその定義域Ｄに繰り込む。その結果、定義域は変化するが、定義域のその変化を通じて関数Ｆは同一であるとみなされる。そのとき、自己言及文 F (F) が得ら れることになる。他方、ウィトゲンシュタインの言い分は、定義域が変化したのだからもはやその関数は同一ではない、というものである。それゆえ、一見自己言及文に見えるものは、実は F’(F) であり、けっして自己言及文ではない。

> 三・三三三　二つの関数に共通なのは文字「Ｆ」にすぎない。だが文字はそれ自体では何も表わさない。

文字が名として意味をもつのは、それが像として用いられ、像としての論理形式をもつからでしかない。いま問題にしているのは述語の論理形式であ

る。そして述語を関数として捉えるとき、その定義域と値域はまさにその述語の論理形式を示すものとなる。それゆえ、ここにおいて定義域は関数の意味に本質的に関わっているのである。それにもかかわらず異なる定義域を通じて同一の関数を想定しようとする態度は、根本的な錯誤として排されねばならない。

ラッセルの述語「 w (x) （x は自分自身に述語づけられない述語である)」は与えられた定義域の上で健全に働きうる。しかし、w 自身をその定義域に繰り込むとき、その定義域は変化し、それはもはや同一の命題関数 w ではなく、別の命題関数 w ' となる。それゆえ、「 w (w) かつ（ w (w) ではない)」という矛盾も生じない。それはただ、「 w'(w) かつ（ w (w) ではない)」にすぎないのである。

三・三三三　かくして、ラッセルのパラドクスは片づく。

資料２

8–2　『論考』の根本思想のありか

ウィトゲンシュタイン自らが『論考』の「根本思想」と称する（四・〇三一二）主張がある。一見すると「根本思想」と呼ぶほどのものには見えないかもしれない。こうである。

論理語は名ではない。

ウィトゲンシュタインは「ではない」「かつ」「または」といった論理語を、対象を表わす名ではなく、要素命題に対する、あるいは要素命題の真理領域に対する「操作」として捉える。それゆえ、ここでわれわれに課された課題は、「操作」というこの概念が『論考』の「根本思想」に関わる中心概念であるのはなぜかを理解することである。そしてそれは、必ずしも見てとりやすいことではない。しかし、ここをきちんと捉えないかぎりは、『論考』を的確に捉えたことにはならない。

まず、「操作」という概念がもつ積極的な側面を検討する前に、「論理語は名ではない」という否定的な主張の意味するところを押さえておこう。

論理語が名ではないならば、論理語は対象を表わさない。それゆえ、論理語に関わる理論は世界のあり方についての理論ではない。これがウィトゲンシュタインのポイントである。たとえば「パンダ」という語はパンダという種を表わしている。それゆえ、パンダについての研究が世界のあり方を調べる科学として成立する。あるいは　「ウィトゲンシュタイン」という固有名はウィトゲンシュタインという対象を表わす。それゆえ、そこにはウィトゲンシュタインという対象のあり方についての探求が成り立つ。「ウィトゲンシュタインは第一次大戦に従軍した」「ウィトゲンシュタインは一九五一年六二歳で死んだ」等々、「ウィトゲンシュタイン」という名を用いた真なる命題

が集められれば、それがウィトゲンシュタインについてのわれわれの知識を表わしている。他方、論理語についての理論はそのような意味で世界のあり方についての知識を形成しない。論理学は自然科学とは異なるのである。論理学のこの独特な性格が、「論理語は名ではない」という一言に集約される。だからこそ、これはウィトゲンシュタインにとって根本思想となった。

しかし、これだけではまだ「根本思想」と呼びうるほどの重みをまったく捉えきれていない。論理語は操作を表わす。このことの意味を見ていくことにしよう。そしてその検討によって、ウィトゲンシュタインがフレーゲ的な論理観とは根本的に異なる論理観を提出していることが明らかになるだろう。

ここに姿を現わすのは、「無限」である。「操作」という概念の重要性は、なによりも無限に関わっている (31)。ウィトゲンシュタインは次のように述べる。

> 五・二五二三　操作の反復適用という概念は「以下同様」という概念に等しい。

実に、この点にこそ核心がある。多少誇張気味に言うことを許していただくならば、これは『論考』全体にとって核心を成すものであると私は考えている。

「以下同様」という句は無限に関わる。われわれが無限に関わるときの唯一のルートがこれである。たとえば自然数を学ぶとき、「以下同様」という言葉がどこかで有効にならなければ、われわれは無限にある自然数を把握することなどできはしない。百まで、あるいは必要ならそれ以上、子どもにひとつひとつ具体的に数えてみせ、　「あとはこんなもんだ」と突き放す。そのとき、「どんなもんだって？　分かんないよ。もう少し教えてよ」と、その子どもがいつまでたってもそう尋ね返してくるようでは、自然数を学ばせることはできない。大人が「以下同様」と見切り、子どもが　「そうか、以下、同様なんだ」とうなづき、そして実際その子どもが以下同様に続けていけるということが、自然数のような無限へと開かれたステップの学習には不可欠のものとなる。

そして、「以下同様」という言葉に実質を与えるのが、操作の反復適用なのである。自然数の場合で言えば「１を足す」という操作を繰り返すこと。「次の数に進みたければ、君がいま数えた数に１を足しなさい。」こうして、いつまででもその操作を続けていける。そう確信したとき、われわれは「自然数を理解した」と言える。そして、それ以外の仕方で無限にある自然数を理解することはできない。

『論考』の目標を思い出そう。思考の限界を捉えること。しかし、思考の限界を思考することはできない。思考の限界に立つことは思考しえぬ領域をも考えることを要求するが、それは不可能である。それゆえわれわれはあくまでも思考可能性の内側に立ち、そこから思考の限界を画定しなければならない。これは、いまの議論の脈絡から言えば、自然数を把握するにも似たこ

とではないだろうか。われわれは自然数の果て、最後の自然数などというものに出会うことはできない。いわば、自然数を内側から捉えなければならないのである。このような場合、つまり、「よし、これですべてを尽くした」という言葉で限界を画定することができない場合、われわれはどこかで「以下同様」という言葉に出会わなければならない。

> 「. . . . . .」という記号で表わされる「以下同様」という概念はもっとも重要な概念のひとつである。(『草稿』一九一六年一一月二一日)

8–3　操作と関数を混同してはいけない（五・二五）

「操作」という概念を「関数」と対比しよう。ここにおいて『論考』のきわめて反フレーゲ的な性格があらわになる。すでに触れたようにフレーゲは関数論的視点から現代論理学を創始した。しかし、ウィトゲンシュタインはそれとはまったく異なる論理観に到達したのである。問題は、関数という概念を用いて「以下同様」という句に実質を与えることができるか、という点にある。「できない」というのがウィトゲンシュタインの答えであったが、「できる」という答えがまさにラッセルのパラドクスを発生させたのである。

フレーゲは、たとえば「犬」という概念を「x は犬である」という命題関数として捉えた。それは個体から真偽への関数とされる。そして関数をも対象化し、それを再び関数に入力することが許されるとするならば、「関数の関数」という考えに至ることになる。そのとき、関数の関数もまた対象化されて関数に入力されうるだろうから、「〈関数の関数〉の関数」が作られ、「以下同様」となる。

カントールが為していた無限集合論の側から述べなおしてみよう。「x は犬である」という命題関数は、それを満たすような対象の集合、つまりすべての犬たちから成る集合を規定する。そこで、そうして作られた集合もまたひとつの対象とみなされる。そうして集合の集合を考えることができる。無限集合論の最大のポイントは、集合を対象とみなすことによって、新たな対象を次々に作っていけるところにある。ふつうに対象とみなされるものの全体として集合を考え、その集合たちの全体として集合の集合を考える。さらにその集合の集合全体として、……「以下同様」。この手順が、カントールの無限集合論の生命線であった。この方法を駆使することによって始めて、カントールは連続体としての実数に到達しえたのである。

しかし、先にも見たように、ここにラッセルのパラドクスが発生した。関数の関数、および集合の集合を、無制限に認めるわけにはいかない。関数の関数や集合の集合を野放図に認めることは矛盾を含むのである。

無限集合をひとつの対象とみなし、さらに無限集合の集合を作っていこうとする態度、しかもそれが無制限に許されるとする態度は、無限に対する実

在論的態度にほかならない。そしてウィトゲンシュタインは『論考』の頃から一貫して無限に対して反実在論的 - 構成主義的態度をとっていたと私には思われる。ウィトゲンシュタインは根本的にカントールのような実在論的態度を認めないのである。ただし、『論考』において集合および集合論について述べた箇所は少ない。次のような主張が、議論なしにいきなり断言されているだけである。

> 六・〇三一　集合論は数学ではまったくよけいである。

ここで、ひとつのエピソードを紹介してみよう。ウィトゲンシュタインが哲学から離れていた一九二八年、ウィーンにおいて、ブラウアーが「数学・科学・言語」という題で講演をし、ウィトゲンシュタインはその講演を聴いた。それだけではない。同席していたファイグルの証言によると、講演後のコーヒー・ショップでウィトゲンシュタインは「異常なまでに雄弁になり、そして彼の後期の著述の発端となった構想の概要を語り始めた。……その夜がウィトゲンシュタインの強力な哲学的関心と活動への復帰を記すこととなった」、というのである。

ブラウアーは直観主義を提唱したオランダの数学者である。そして直観主義とは、無限に対する構成主義をもっとも強く押し出す立場にほかならない。その講演を聴いてウィトゲンシュタインはどうしてそれほど興奮したのか。ありそうにないのが、その講演によってウィトゲンシュタインが構成主義に目を開かされたというものである。ブラウアーの講演を聴いて目からウロコが落ち、『論考』の立場を反省して後期の哲学へと踏み出し始めた―　というのは、ストーリーとしても単純すぎる。おそらく、その講演に『論考』の精神に通じるものを見てとりつつも、だからこそそこに嗅ぎとられた違和感におおいにかきたてられたのではないだろうか。似ている。しかし、はっきりした違和感がある。この違和感は何か。それがウィトゲンシュタインをファイグルたちに対して多弁にした。さらに無責任に言うならば、フレーゲやラッセルの実在論のもとで構成主義を打ち出したのが『論考』であったとすれば、構成主義をも解体し始めたのが、中期から後期の彼の哲学ではなかっただろうか。

話を戻そう。パラドクスの発生地点は、「集合の集合、〈集合の集合〉の集合、……以下同様」と進み、あるいは「関数の関数、〈関数の関数〉の関数、……以下同様」と進むところにある。ラッセルのパラドクスが集合に対しても関数に対してもまったく同型の二つのヴァージョンをもっていたように、両者は同じ構造をもっているように思われる。しかし、実はそうではなかった。無限集合論に対応するものを関数で生み出そうとすることは、関数の本性を決定的に見まちがえているのである。命題関数は集合と同等のものではない。『論考』は集合論をよけいなものと断罪したが、関数をよけいだとは言わない。関数は集合よりもずっと健全な道具立てでありうるのである。

第４章でラッセルのパラドクスを論じたときの議論を思い出していただきたい。　『論考』は関数をたんに入出力の対照表にすぎないものとみなした。つまり、定義域が異なれば関数は関数として異なるのである。これはいまの議論の脈絡で何を意味するだろうか。命題関数を考えてみよう。フレーゲの規定に従えば、それは個体から真偽への関数である。そこで、動物を定義域とする命題関数「ｘは犬である」を考えよう。それは動物の集合という定義域の上で働き、犬であるものと犬でないものとを弁別する。こうして、この命題関数は動物という集合から犬という部分集合を取り出す。そのかぎりにおいて、すなわち、あくまでも定義域となる集合の部分集合を取り出すかぎりにおいて、命題関数は集合に等しい。つまり、命題関数は定義域の集合の部分集合を取り出すものでしかないのである。それは与えられた定義域よりも小さい集合を取り出してくるにすぎない。命題関数とは部分を切り分ける弁別規則であり、出発点を越えて何ものかを構成していく力をもった構成規則ではないのである。

たとえば関数 ｆ (x) ＝ｘ＋１ を考えよう。一見するとこの関数を用いて「以下同 様」という地点に立てるようにも思われる。つまり、こうである。最初の入力を０とする。そのとき出力は１である。次にそれを入力する。そうすると新たな出力２が得られる。次にそれを入力する。そうすると新たな出力３が得られる。以下同様。かくして、０から出発し関数 ｘ＋１ を用いて自然数が得られる。しかし、ウィトゲンシュタインの観点からすればこれはまったくの誤解である。関数は定義域とコミになってのみ意味を確定する。それゆえもし定義域が最初｛０｝であるならば、それを関数 ｘ＋１ に入力して得られた１はもうその関数に入力できない。出力１が再び関数 ｘ＋１ に入力されるためには最初から定義域に１が含まれていなければならない。 １を入力して得られた出力２に関しても同様である。その２をさらに関数 ｘ＋１ に入力させるためには、最初から定義域に２が入っていなければならない。それゆえ、関数 ｘ＋１ で自然数を出したいのであれば、最初から定義域が自然数でなければならない。関数 ｘ＋１ には自然数を構成する力はないのである。

それに対して、操作はそうではない。操作を施す相手を「基底」と呼ぶが、操作は基底と独立に定まるのである。操作「＋１」を考えよう。｛０｝を最初の基底として、それに「１を足す」という操作を施す。その結果１が得られるが、この１を再び基底に繰り込むことができる。そして１に対しても「１を加える」という操作を施す。その結果２が得られ、それもまた基底に繰り込むことができる。操作の結果は次々と新しく基底に繰り込むことができる。

「二つに折る」という操作を考えてみよう。一枚の紙が与えられる。それを二つに折る。すると目の前には新しく二つに折られた紙が現われる。そこでそれをまた二つに折る。そうするとこんどは四つに折られた紙が現われる。それをまた二つに折る。以下同様。何をもってきても、「二つに折る」という

操作はひとつである。新聞紙をもってくる。「二つに折る」。Ｂ５のコピー用紙をもってくる。「二つに折る」。葉書をもってくる。「二つに折る」。操作の基底は関数の定義域とはまったく異なる。関数は入出力の対照表にすぎず、それゆえ定義域が異なれば異なった関数であるが、操作は基底と独立に同一性をもっている。だからこそ、操作は無限列を構成しうるのである。