

# 東工大文芸部『論理哲学論考』読書会 第5回

伊澤侑祐

平成 28 年 1 月 4 日

## 1 序

### 範囲

『論理哲学論考』（刊：岩波文庫）4.2 — 5.5423

### 進行表

- 予定の確認
- 該当範囲の読み合せ（今回はブレインストーミングは省略）<sup>1</sup>
- 前回まで読書会の振り返り
- 内容確認
- レジューメ読み合せ
- 課題についての議論<sup>2</sup>
- まとめ
- 次回の予定の確認

### 1.1 概要

今回は操作の繰り返しとしての「無限」という概念に触れた段階で終わった。今回は、その続きから始まることになる。今回のメインテーマは「真理操作」である。真理操作とは、要素命題からどのようにして複合命題を生成するのか、その過程を見ていくことになる。

---

<sup>1</sup>今回は主に知識の整理が目的なので、読み合せでは内容の確認にとどめる

<sup>2</sup>なお、今回は「言ってしまうと当たり前」のような内容が多く、内容の発展的な議論をするよりは「疑問点の確認」に時間を当てようと思う

## 2 本文

### 2.1 分析

4.22 — 命題を分析していけば、その結果は明らかに、名が直接結合してできた各要素命題でなければならない。要素命題が存在するからこそ、いかにして命題と命題の結合がなされるものか問題になるのである。

4.23 — すべての真な命題要素の列挙によって、世界は完全に記述される。世界は、すべての要素命題をあげ、さらにどれ真でどれが偽かを付け加えれば、完全に記述しうる。

本章のはじめには分析について書かれている。分析とは、分解と構成の繰り返しである。これを簡単に言うと「バラして組み合わせる」ことである。バラし方にもルールがあって、それは論理形式に則っていなければならないということである。

また、ここにある「論理語」とは「かつ」「または」「よって」という接続詞であり、名ではない。これについては後ほど確認する。

#### 2.1.1 分解のプロセス

- 第一段階：  
命題の検証や推論といった言語実践の中で、有意味／無意味／ナンセンスを弁別する我々の”言語直感”を頼りに分析がなされ、要素命題と論理語が区別される。
- 第二段階：  
要素命題は名と対象の対に分解される。そして名は、論理形式によってどのような配列が可能であるかチェックされ、対象も論理形式によっていかなる事態の構成要素となるかチェックされた上で分解される。

#### 2.1.2 構成のプロセス

- 第一段階：  
名の論理形式にしたがって可能な要素命題のすべてが構成される。構成された要素命題は、すべての可能な事態を表現するものとなる。
- 第二段階：  
事態の集合として状況が作られ、可能な状況の全体として論理空間が貼られる。また、命題では論理語によって要素命題から複合命題が作られる。

### 2.2 論理記号

4.0312 — 「論理定項」は何ら対象の代わりをするものでない

とあるように、記号「 $\sim$ 」「 $\vee$ 」「 $\cdot$ 」「 $\supset$ 」は論理語（論理定項）といい、名ではなく、対象でもない。つまり「操作」である。各操作の真偽は 4.3 のような真理値表に従う。

- 否定  $\sim p$  — 「 $p$  でない」

- 論理和  $p \vee q$  — 「p または q」
- 論理積  $p \cdot q$  — 「p かつ q」  $p \wedge q$  と同じ意味<sup>3</sup>（カッコ（ ）の代用として捉えると良い）
- 条件法  $p \supset q$  「p ならば q」
- $p | q$  — 「p でも q でもない」
- $(x).fx$  — 「すべては f である」または「すべての x に対して x は f である」
- $(\exists x).fx$  — 「f であるものが存在する」または「ある x に対して、x は f である」

推論関係が包含関係で捉えられるのは、私たちにとっては非常に馴染みのある事実であろう。

## 2.3 命題

真理条件について知識を確認するために、本文を参照しよう。

- 4.3 — 要素命題の真理可能性は、事態の成立・不成立の可能性を意味している。
- 4.41 — 要素命題の真理可能性が命題の真偽の条件である。
- 4.431 — 要素命題の真理可能性との一致及び不一致の表現が、それぞれその命題の真理条件を表している。

つまりここで大事なものは「真理可能性との一致 / 不一致」である。また、真理条件の中には「トートロジー」「矛盾」という極端なものがある。

トートロジーは恒真命題、つまり全命題の内側の極限点、矛盾とは恒偽命題、つまり全命題の外側の極限点である。

## 2.4 論理語、操作、そして無限

論理語（論理定項）が名でないことを考察しよう。

論理語が名でないならば、論理語は対象を現さない。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。「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、深く調べることができる。「ウィトゲンシュタインは第一次世界大戦に従軍した」「ウィトゲンシュタインは 1951 年 62 歳で死んだ」など、「ウィトゲンシュタイン」という名を用いた真なる情報を集めれば「ウィトゲンシュタイン」について分析できる。<sup>4</sup>

一方で、論理語はどうかというとそうではない。「よって」「または」などの真なる命題とはそもそも何なのか規定しうることも叶わない。したがって、論理語は名でないと結論を出すことができる。

何より、「操作」という概念自体が「無限」に深く関わっている。

- 5.2523 — 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。

<sup>3</sup> ちなみに Haskell では関数合成をドット  $\cdot$  で表現する。Haskell では右から評価されるが、\$ でつなぐことによって左から評価されるようになる）

<sup>4</sup> 野矢先生の本の本文では知識を形成するとある

「以下同様」の「以下」に無限が隠れていることがわかるだろうか。「以下」には範囲が規定されていないが、これからさきもずっと「同様」であることがひと目でわかるだろう。これが「無限」なのである。

論考の目標は思考の限界を捉えることである。しかし、思考の限界を思考することこそ不可能である。だからこそ、思考の限界を画定するために「無限」を用いないといけない局面が必ず訪れるに違いない。

ここから今回の内容

## 2.5 操作と関数

「操作」という概念を「関数」と比較する。両者は似たような働きを持ち、同じような概念に見えるものの、実際は深いところでかなり違っている概念である。フレーゲは関数論的な立場から現代論理学を創始したが、それは「ラッセルのパラドクス」という非常に重大な問題を孕んでいることをウィトゲンシュタインに指摘された。なぜパラドクスが発生し、そしてなぜラッセルたちはそのパラドクスに気付けなかったのか、それは「無限」に対する論理観が根本的に違っていたからである。問題は、関数という概念を用いて「無限」を表現できるか、という点にある。

フレーゲは関数を対象化し、それを再び関数に入力することが許されるとして「関数の関数」が許されるとした。「関数の関数」も対象なので、それを再び関数に入力して「関数の関数の関数」を生成して、再帰的に関数を作ることができる。そしてやがて無限に到達するのである。

これをカントールの無限集合論の側から述べ直すならば、一つ集合を規定し、それは対象とみなされるのでさらに集合の集合を作る。こうして新たな対象を次々と作っていく。普通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、こうやって抽象度を上げていき、カントールは連続体としての実数に到達した。

しかし、ウィトゲンシュタインはこれにまったをかけた。ここでラッセルのパラドクスが出てくるのである。では一体何が問題であったのだろうか。

### Q.1 なぜ関数を無限回適用して無限を構成してはいけないのだろうか？

これに対し野矢茂樹氏はこう述べる。

無限集合を一つの対象とみなし、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、無限に対する實在論的態度に他ならない。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「無限は操作の反復としてのみ構成される」という半實在論的-構成主義的態度を取っていたと私には思われる。

確かにそうである。無限とは扱えるものであるとする前提がそもそも間違っている。まず、私たちは無限を知覚できない。「非常に多くの何か」「数え切れない何か」を想像することはたやすいが、それが無限であるかどうか判定することはできない。したがって、無限とは無限とは實在的なものではなく、操作によって構成されるものである。そもそも、無限集合論に対応するものを関数で生み出そうとする前提から間違っており、関数は単なる入出力の対応表に過ぎないのであった。

真理関数の説明に戻るが、これは関数ではない。なぜなら、真理関数において入力使命題であり、変項ではないからである。そして、操作は真理領域（すなわち Boolean）を取り出すことに他ならない。もっとも、ウィトゲンシュタインは次のような注意を与えていた。

5.44 — 真理関数は実質的な関数ではない。

これで操作と関数の違いについて意識的になれるだろう。

### 3 命題の構成について

論理学に立ち入ることによって、論理空間がどのようにして張られるのかより理解が深まる。「論考」は論理語を否定詞と接続詞に限定している。これは現代論理学の観点から「命題論理」と呼ばれる道具の体系化で止まっており、表現力に関しては述語論理に比べて乏しいと言わざるをえない。

ここで、量子化について少し触れておこう。量子化とは、命題論理の論理語に加えて、量に関する論理語「すべて」を持つものである。ここでまず「論考」がどのように量子化を扱ったか押さえる。

点灯論理空間を考える。

$w_1 \cdots \phi$

$w_2 \cdots a$  - 点灯

$w_3 \cdots b$  - 点灯

$w_4 \cdots a$  - 点灯、 $b$  - 点灯

命題は論理空間の中にそれが真になる領域を規定する。命題「a」は  $\{w_2, w_4\}$  において真、命題「b」は  $\{w_3, w_4\}$  において真となる。

「真理操作」とは、命題に対して論理語を適用する（言葉を被せる、という言い方はどうだろうか）ことによって真となる部分を取り出す操作である。例えば、命題「a」の否定をとると  $\{w_1, w_3\}$  を取り出せる。

これは状態が有限個しかない。それには理由があり、次に「有限回」とはっきり述べられているからである。

5.32 — すべての真理関数は、要素命題に対して真理操作を有限回繰り返し適用することによって得られる。

しかし、これは少し怪しい。検証するために、論理語について考察しよう。

すべての論理語は否定

$$N(\bar{\xi})$$

を用いて定義できる。

- 「p ではない、かつ、q ではない」 —  $N(p, q)$
- 「p かつ q」 —  $N(N(p), N(q))$
- 「p または q」 —  $N(N(p, q))$

ではココで疑問が生じる。すべての論理語を否定によって構成できてしまった。無限は構成するものである<sup>5</sup>とは先ほど触れたばかりであるが、

Q.2 果たして真理関数は、要素命題に真理操作を無限に繰り返し適用できてしまうのでは？

これは命題 5.32 によって直ちに却下されてしまう。真理関数が無限個生成されることは至極当たり前に感じてしまうが、それでも我々は命題 5.32 に従うべきだろうか。

私は、命題 5.32 はウィトゲンシュタインの苦肉の策であったと思わざるをえない。哲学的な立場では無限個の「かつ」「または」は許されている。無限のセクションでやったように、「以下同様」としてしまえばいいだろう。ここは少し議論を要する命題だろう。

ともあれ、これで基本となる道具は揃った。論理空間における操作として真理操作が導入され、その真理操作にしたがって要素命題が複合命題へ構成されていく。これが「語りうるもの」のすべてである。

## 4 余談

### 4.1 論理のア・プリオリ性

本範囲では「真理操作」がメインテーマに据えられているため、その性質についても軽く触れておこう。

「ア・プリオリ」とは、検証に先立ち、検証を可能にするために前提とされていることである。本文においては、ア・プリオリ性について幾つか述べられている。命題 5.133 において「すべての導出はア・プリオリに成立している。」とあるが、これは操作によって導出される命題は経験に先立って「成立している」としなさい、と言っているのである。

次の命題は色々な意味で押さえておきたい。

5.4731 — 自明性など論理においては不要でしかない。論理がア・プリオリだというのは、論理に反しては思考不可能ということに他ならない。

この命題について少し考えてみよう。

Q.3 論理に反しては思考不可能というのは、一体どういうことなのだろうか？

命題 7「語りえぬものについては、沈黙しなければならない」とあり、論考はそこで筆を置いている。この叙述もまた面白いもので、「語りえぬもの」はそもそも非論理的なので、私たちは論理を以ってそれを思考することができない。つまり、論理に反することこそが「沈黙しなければならない」ことなのであると私は考えるがいかがだろうか。

### 4.2 真理関数とプログラム

真理関数とは真偽を入力して真偽を返す関数のことである。イメージとしてはプログラムにおける Boolean と捉えると良い。むしろ、真理関数はプログラム言語としてみると大まかなイメージは理解出来よう。たとえば、

<sup>5</sup>「以下同様」と宣言すること、すなわち繰り返しの操作を構成することと言い換えた結果、このような言葉遣いになった

```

1 < 2          // == true
4 < 2          // == false
"abc" > "deg"  // == false
"a" == "a"     // == true

```

のような表現としてみることができる。しかし、この説明はやや間違っている。なぜなら、論考においては真理関数という用語は命題を否定したり、命題と命題を「かつ」「または」「ならば」などで接続したりして作られる複合的な命題ないしその形式だからである。では、より計算機との関連の中で真理関数について考えてみよう。

Q.4 真理関数とは計算機において一体どのようなものなのだろうか？

私としては、真理関数とは様々な演算の組み合わせによって得られるアルゴリズムのようなものだと思っている。計算機は論理学の発展とともにあった。今私たちが用いている様々なデバイスは、高度な計算によって動いている。その計算というのは、単なる値の演算だけではなく、論理演算やデータ処理も含まれている。その処理というのは全て論理をベースに行われているので、真理関数とはまさにプログラムの原型となったものなのではないかと考える。

### 4.3 否定/論理積/論理和に関連する命題

否定/論理積/論理和について原文を押さえておく。

#### 4.3.1 否定

5.5 — いかなる真理関数も、要素命題に次の操作を反復適用した結果である。

$$(\neg \neg \neg \neg T)(\xi, \dots)$$

これは下の括弧内を全て否定したものであり、私はこの操作をこれら諸命題の否定と呼ぶ。

これは代わりに  $N(\xi)$  と記す。さらに、否定の用法として、二重否定は肯定 ( $\neg\neg p = p$ ) であるとも述べられている。

#### 4.3.2 論理和/論理積

5.513 —  $p$  と  $q$  を共に肯定する全てのシンボルに共通なもの、それが命題

$$p \cdot q (p \wedge q)$$

である。 $p$  か  $q$  いずれかを肯定する全てのシンボルに共通なもの、それが命題

$$p \vee q (p \vee q)$$

である。

さらに、ナンセンスな表現について命題 5.5351 が取り上げている。

## 5 課題

今まで登場した課題を整理しよう。

Q.1 なぜ関数を無限回適用して無限を構成してはいけないのだろうか？

Q.2 果たして真理関数は、要素命題に真理操作を無限に繰り返し適用できてしまうのでは？

Q.3 論理に反しては思考不可能というのは、一体どういうことなのだろうか？

Q.4 真理関数とは計算機において一体どのようなものなのだろうか？

以上の4題を今回の発展課題とするが、時間のない場合、Q.4は飛ばしてQ.2 Q.3 Q.1の順番で処理してほしい。

## 参考文献

- [1] 論理哲学論考, ウィトゲンシュタイン (訳: 野矢茂樹), 岩波文庫, 2003
- [2] 論理哲学論考, ウィトゲンシュタイン (訳: 丘沢静也), 光文社古典新訳文庫, 2014
- [3] ウィトゲンシュタイン『論理哲学論考』を読む, 野矢茂樹, ちくま学芸文庫, 2002