哲学演習
「論理学入門」
第 2 回

池田 真治*

2014 年 4 月 22 日

「判断、概念等の間に成立する論理的事態は、その解釈が日常言語の表現においては容易に生じうる不明瞭さを含まないような、定式によって表されることになる。推論に見られるような、論理的帰結への移りゆきも、一番もとの要素にまで分析するならば、代数学における計算規則に類似したある規則にしたがって、最初の式を形式的に変形してゆくことであることがわかる。こうして論理的思考は、論理計算にその姿を写すことになるのである。純粋に内容的な思考では原理的に解決不可能な諸問題を、効果的に取り扱うことのできる道がこの計算によって開かれる。」——ヒルベルト、アッケルマン『記号論理学の基礎』序より。

「論理学を可能にしているものは、われわれの精神の内にある一般的思念の存在である——すなわち、クラスについて把握するわれわれの能力、および個体的要素をある共通な名前によって指定するわれわれの能力である。こうして論理学の理論は、言語と密接な結びつきを持つ。」——G. ブール『論理学の数学的分析』序論より。

目次

* 富山大学 人文学部 shinji@hmt.u-toyama.ac.jp ; URL: http://researchmap.jp/shinjike

2 論理学の言語

2.1 自然言語と人工言語はどう違うのか？

自然言語（natural language）・・・われわれが普段使っている日常言語。つまり、ここでの日本語。人間が意思疎通を目的に、自然に発展・形成してきたもの。

自然言語は、表現が極めて豊かであるが、その分、命題の論理形式（logical form）にだけ注目するのには、向いていない。

論証の妥当性（validity）だけが問題になる場合、論証の構造を明確にするのに特化した言語の方が都合がよい。そこで、形式的な推論を扱うのに適した、人工言語をわれわれは作ることになる。

人工言語（artificial language）・・・言語規則が人為的なしかたで、明確に規定されている言語。数式、コンピュータのプログラム言語（Lisp、C 言語、java, etc.）や記号論理学の言語など。将来の自然言語となるべく、国際共通語ないし普遍言語を目指してつくられた人工言語（エスペラント語やロジバンなど）もある。

われわれはこれから、自然言語から抽象して、人工言語を作る。

2.2 論理学の目標

(1) 論理学の目標は、命題の真理と区別された、論証の妥当性とは何かをはっきりさせることにある。

(2) そのためには、論証の内容はいったん忘れ、論証の形式に注目する必要がある。

(3) その際「でない」、「または」、「かつ」、「ならば」、「すべて」などの論理定項（logical constants）に注目する必要がある。

(4) 自然言語では命題の論理形式が文法形式に覆い隠されてしまうことがある。

(5) したがって、論証の妥当性にかかわる部分（論理定項）と、そうでない部分が最初

からはっきり分かれているような言語、論証の論理形式が一目でわかるような言語を、人工的につくってしまおう。

2.3　記号化

(1)「でない」、「または」、「かつ」、「ならば」などの論理定項に注目して、与えられた一つの文（複合命題）を、それを構成しているできるだけ小さい文（単純命題）にまで分解する。

(2) そうした原子的な文に記号 P, Q, R, ... を割り当てる。

(3)「または」を「∨」、「かつ」を「∧」、ならばを「→」、「でない」を「¬」で表す。

¶教科書 p.18-20

2.3.1　演習

練習問題 2 をやってみよう。

2.4　正しい論証にかかわる情報

・and と but のニュアンスなど、自然言語には相手に伝えたい情報がある際、微妙な意味の違いを表現する言葉があるが、これから作る人工言語では考えない。

・なぜなら、それらは、正しい論証にかかわる情報ではないからだ。

・論理学の目標に照らすと、正しい論証にかかわるのは、真偽という評価軸である。

2.5　真理関数という考え方

・「または」、「かつ」、「でない」、「ならば」を、真理関数型結合子と呼ぶ。

・重要なのは、真理関数という名前よりも、真理関数という考え方だ。

・それは、単純命題の真偽の値が決まれば、全体の複合命題の真偽の値も決まる、というもの。

2.5.1　例

「富山は今日晴れていない」は、「富山は今日晴れている」が真のとき、偽になる。つまり、「でない」は、真な単純命題を偽にする、真理関数である。

2.6　人工言語 L

これからつくる論理学を記述する人工言語（形式言語）を **L** と呼ぶ。それは、誰もがそれを使って論証することができる言語であるべきである。

論理学は普遍的な研究である。すなわち、（人間には実際にできないようなものまで含む）ありとあらゆる論証、ありとあらゆる論理式について当てはまる定義ができ、無限に多くあるすべての論証や論理式について成り立つ事柄を研究する。

言い換えると、論理学は、原理的に可能な推論・論理式の全体という抽象的な対象の全体を相手にする。

そのために、人工言語 **L** は、

(1) 論理式であることの基準が曖昧であってはならない、かつ、

(2) すべての論理 s 機にうちて様々な事実を証明することを可能にしてくれるものであることが望ましい。

2.6.1　人工言語 L の定義

L の語彙

(1) 原子式
   $P, Q, R, \cdots$
   $P_1, P_2, P_3, \cdots$

(2) 結合子
   $\rightarrow, \vee, \wedge, \neg$

(3) 補助記号
   (,)

定義 1 (論理式の定義)**.**
(1) 原子式すなわち $P, Q, R, \cdots, P_1, P_2, P_3, \cdots$ は論理式である。
(2) A, B を論理式とすると、$(A \wedge B), (A \vee B), (A \rightarrow B), (\neg A)$ はおのおの論理式である。
(3) (1), (2) によって論理式とされるもののみが論理式である。

・このような定義の仕方を、帰納的定義(inductive definition) とか回帰的定義(recursive definition) と呼ぶ。

・A, B は、特定かつ具体的な論理式ではなく、無数の論理式を代表している。A, B には論理式であれば、何を入れてもかまわない。A, B は **L** の語彙ではないが、**L** でつくられた論理式をそこに入れられる文字である。こうした A, B をメタ論理的定項 (meta-logical variable) とか図式文字 (schematic letter) と呼ぶ。（まぁ、とくに名前は覚える必要もない。）

2.6.2　形成木 (formation tree)

¶教科書 p.26

2.6.3　演習

練習問題 3 をやってみよう。

2.7　帰納法による証明

2.7.1　帰納的定義の意味

帰納法は数学っぽくて難しい！ と思うかもしれない。今は、はっきりわからなくとも、後でわかるようになればよい。

定義 2 (数学的帰納法)**.** P がある命題ないし性質で、k, n が自然数のとき、

$$\frac{\text{「}P(1)\text{ が成り立つ」} \land \text{「}P(k)\text{ が成り立つ} \rightarrow P(k+1)\text{ が成り立つ」}}{\text{「すべての } n \text{ について } P(n)\text{ が成り立つ」}}$$

2.7.2　数学的帰納法による証明

¶教科書 p.27

板書で説明します。

2.7.3　式の長さについての帰納法による証明

なぜ帰納法について説明していたのか、というと、われわれは先ほど論理式を帰納的に定義したので、論理式一般について成り立つ性質 P について証明したい場合、帰納法が使えるからだ。

定義 3 (式の長さについての帰納法)**.**
P がある命題ないし性質で、A, B が論理式のとき、

$$\frac{\text{「原子式 について } P \text{ が成り立つ」} \\ \wedge \\ \text{「}A, B \text{ について } P \text{ が成り立つ} \rightarrow \neg A,\ A \vee B,\ A \wedge B,\ A \rightarrow B \text{ についても } P \text{ が成り立つ」}}{\text{「すべての論理式について } P \text{ が成り立つ」}}$$

定理 2.1. すべての論理式は同じ個数の右カッコと左カッコを持つ。

Proof.「同じ個数の右カッコと左カッコを持つ」を P と置いて、さっきに定義した、式の長さについての帰納法を適用すればよい。(詳しくは板書で) □

「すべての論理式について P であることを示しなさい」という一般的問題には、このように、式の長さについての帰納法、あるいはそれと実質的に同じことをやっている、結合子の数についての帰納法が使えるということである。

定義 4 (結合子の数についての帰納法)**.**
P がある命題ないし性質で、A, B, C が論理式のとき、

$$\frac{\text{「}A \text{ が } 0 \text{ 個の結合子を含むとき、つまり } A \text{ が 原子式 のとき、} P \text{ が成り立つ」} \\ \wedge \\ \text{「}k \text{ 個以下の結合子を含むすべての論理式について } P \text{ が成り立つとする。} \\ \text{このとき、} k+1 \text{ 個の結合子を含む論理式、} \\ \text{つまり } \neg B,\ B \vee C,\ B \wedge C,\ B \rightarrow C \text{ についても } P \text{ が成り立つ」}}{\text{「すべての論理式について } P \text{ が成り立つ」}}$$

2.8 Unique Readability Theorem

Unique Readability Theorem とは、「論理式がそれぞれただ一通りにしか読めない」ということを主張した定理である。

それを、式の長さについての帰納法によって証明する。

定義 5 (始切片)**.**
論理式を構成する（有限の）記号列を考える。論理式 A の左端から順番にとっていき、途中で切り落とした記号列を始切片という。ただし A そのものは始切片ではないとする。
より厳密には、論理式 A を構成する記号列を $S = \langle x_1, x_2, ..., x_k, ..., x_n \rangle$ としたとき、S の $1 \leq k < n$ で表される真な部分列 $\langle x_1, x_2, ..., x_k \rangle$ を A の始切片という。

問題. $(A \to B)$ の始切片をすべて挙げよ。

すべての論理式のすべての始切片は、論理式ではないことが言える。

定理 2.2. すべての論理式 A について、A のどの始切片も、右カッコより多数の左カッコを含んでいる。

Proof. 証明には、式の構成についての帰納法を使う。（詳しくは板書で） □

問題. 教科書 p.31 の、定理 2 の証明で省かれている、$(A \wedge B), (A \to B)$ のケースについての証明を埋めよ。

定理 2.3. すべての論理式 A について、A のどの始切片も、論理式ではない。

Proof. 定理 2.1 と定理 2.2 より明らか。なぜなら、始切片ならば左右のカッコの数が違うが、論理式ならば左右のカッコが同じでなければならないからである。 □

Unique Readability Theorem のモチベーションは次のようなものだ。われわれが作ろうとしている論理学において、「論理式が、ただ一通りにしか読めない」という性質を持っていることが重要である。

なぜなら、われわれは、いろいろな読み方ができてしまうような紛らわしい論理式を、排除したいからだ。たとえば、$(A \wedge B \vee C)$ などと書いてしまうと、$((A \wedge B) \vee C)$ なのか $(A \wedge (B \vee C))$ なのか、分からない。実際にこれらは違うものなので、このような曖昧性を許してしまうと、正しい論証はできなくなってしまう。

そして、われわれが定義した論理式は、実際そのように作られているのだが、そのことが正しく成り立っていることを確かめるのが、次の「証明」である。

定理 **2.4** (unique readability theorem)**.**
A, B, C, D はすべて論理式とする。また、△、▲は任意の相異なる 2 項結合子 $(\to, \vee, \wedge)$ のどれかだとする。このとき、
(1) $(A \bigtriangleup B) = (C \blacktriangle D)$ というようなことはない。
(2) $(A \bigtriangleup B) = (\neg C)$ というようなことはない。
(3) $(A \bigtriangleup B) = (C \bigtriangleup D)$ ならば、$A{=}C$ かつ $B{=}D$ である。
ただし、ここでの「=」は、両辺にある記号列が全く同じ記号が同じ順序で並んだ列であることを意味するものとする。

Proof. (1), (2) の証明は背理法による。

(1) 任意の相異なる 2 項結合子△、▲について、$(A \bigtriangleup B) = (C \blacktriangle D)$ が成り立つと仮定する。このとき、たとえば $(A \wedge B) = (C \vee D)$ が成り立っている。左カッコは始切片として共通である。したがって、$A \wedge B) = C \vee D)$ である。このとき $A = C$ でなくてはならない。なぜなら、$A \neq C$ とすると、A, C の一方が他方の始切片でなくてはならなくなるが、先の定理 2.3 より、論理式の始切片は論理式ではありえないからである。よって $A = C$。

すると、$\wedge B) = \vee D)$ となるが、始切片が等しくなるので、$\wedge = \vee$ となるが、これは矛盾。したがって、任意の相異なる 2 項結合子△、▲について、$(A \bigtriangleup B) = (C \blacktriangle D)$ が成り立つという仮定が誤りだった。

(2) についても (1) とほぼ同様のやり方で示せる。

(3) 任意の 2 項結合子△について、$(A \bigtriangleup B) = (C \bigtriangleup D)$ とする。このとき、たとえば、$(A \wedge B) = (C \wedge D)$ である。始切片の左カッコが等しいので、$A \wedge B) = C \wedge D)$ である。すると、さっきと同様の理由により、$A = C$ でなくてはならない（定理 2.3）。よって、$\wedge B) = \wedge D)$。$\wedge$ は等しいから $B = D$ でなくてはならない。以上より、$A = C$ かつ $B = D$ である。したがって、$(A \bigtriangleup B) = (C \bigtriangleup D)$ ならば、$A = C$ かつ $B = D$ である。 □

2.8.1　カッコ省略の規則

(1) 一番外側のカッコについては省略してよいことにする。

(2) $(\neg A)$ の形で現れる論理式については、外側のカッコを省略して単に $\neg A$ と書いてよいことにする。

2.8.2　論理式の形に関するいくつかの用語

¶教科書 p.34 参照。（細かい用語の定義は、その都度カクニンすればよい。）

2.8.3　宿題

練習問題 4 と練習問題 5 をやってくること。