高校数学の論理教材の体系と導出原理に基づく証明技法について

林　　　雄一郎
北海道情報大学

On the System of the Logic Teaching Materials in the High School Math and the Proof Techniques Based on Resolution Principle

Yuuichirou HAYASHI
Hokkaido Information University

平成27年11月

北海道情報大学紀要　第27巻　第 1 号別刷

〈論　文〉

高校数学の論理教材の体系と導出原理に基づく証明技法について

林　　雄　一　郎

On the System of the Logic Teaching Materials in the High School Math and the Proof Techniques Based on Resolution Principle

Yuuichirou　HAYASHI*

要　旨

高校数学で扱われている数理論理学に関連する内容は、「数学Ⅰ」の「数と式」で集合と命題に関する基本概念、「数学A」の「図形の性質」での三角形や円に成り立つ性質の証明、さらに「数学Ⅱ」では「いろいろな式」の「式と証明」で等式や不等式の証明、また「数学B」では「漸化式と数学的帰納法」で漸化式や数学的帰納法を学ぶ基本的概念である。本稿においては、数理論理学の立場からこれらの内容を吟味し、指導の工夫や着想を支える教材研究の背景的知見を整理するとともに、数理論理学の知識、導出原理に基づく自動証明技法やPROLOGの活用例を考察する。これらの考え方や技法は人工知能の基礎分野でもあり、学生の興味・関心を引きつける教材づくりの基礎となるにちがいない。

Abstract

The contents in conjunction with mathematical logic contained in high school mathematics are fundamental concepts to learn set and proposition in the unit of " numbers and expression" of "Math Ⅰ",proof of properties on triangle and circle in the unit of "properties of figures" of "MathA",and proof of equations and inequalities in the unit of "expression and proof" of "MathⅡ",recursion formulas and mathematical induction in the unit of "recursion formula and mathematical induction" of "MathB". In this paper, the author reviews these contents from viewpoint of the mathematical logic and arranges the knowledge to become the background of the study of teaching materials that support devices and ideas for instruction ,and studies the knowledge of mathematical logic and paradigms of automatic proof techniques based on Resolution Principle and PROLOG.These ideas and techniques are basic field of artificial intelligence ,therefore these must be the base to make teaching materials which attract interest of students .

キーワード

論理、論証、証明、命題論理、述語論理、論理体系、形式的証明、Herbrandの定理、モデル、反駁、定理証明、導出原理、論理プログラミング、PROLOG、人工知能

*北海道情報大学情報メディア学部情報メディア学科特任教授　Specially appointed Professor, Department of Information Media ,Faculty of Information Media

1　はじめに

高校数学で扱われている数理論理学に関連する教材内容（以下、"論理教材"という）は次の通りである。

「数学Ⅰ」の「数と式」では、集合と命題に関する基本概念を理解し、それを事象の考察に活用するとしている。

「数学A」の「図形の性質」では、三角形や円に成り立つ性質に関する定理の証明が含まれ、「数学Ⅱ」の「式と証明」でも、記号操作で行う等式・不等式の証明が扱われる。また、「数学B」の「漸化式と数学的帰納法」では、有限の立場から帰納的方法で無限の対象となる数列を定義し、証明する。いずれも証明を構成する論理や証明の形式・方法に対する認識を深めさせ、もって論理的思考力を培うものである。

ところで、授業の成否は扱う教材に対する教師の教材観に大きく依拠している。論理教材の教材観には、数理論理学やこれに関連する数理科学の素養が必要となる。

本稿においては、数理論理学に関連する教材研究の基礎となる知識体系や情報科学においてすでに確立された理論・技術となっている導出原理に基く定理証明に関連する知識を整理するとともに、それに基づく論理プログラミング言語 PROLOG の活用例を考察している。

2　論理教材の内容

高校数学の教科書（高橋他、2011）から論理教材を挙げ考察する。

2－1「数学Ⅰ」

集合と命題に関する基本的な概念を理解し、それを事象の考察に活用する力量を養うとしている。

2-1-1　集合

ここでは集合と要素、有限・無限集合、部分集合、集合の相等、共通部分・和集合、空集合・補集合、集合に関する De Morgan の法則を扱う。集合の概念は、数学を構成するのに使われ、論理とは密接不可分な関係がある。集合を定義するには条件(述語論理式)が使われ、それは集合の性質を規定する。例えば、$A=\{x \mid x$は5以下の自然数$\}$は "xは5以下の自然数である" という条件を満たす集合$\{1,2,3,4,5\}$を表示する。これは Cantor の素朴集合論を公理化した Frege の次の抽出公理に依拠する。

$$\exists A\forall x\left(x\in A\leftrightarrow F\left(x\right)\right).$$

記号$\leftrightarrow$は同値の表示(4・2で定義、6・1で意味の定義)である。この公理は、集合$A=\left\{x \mid F\left(x\right)\right\}$の存在を保証するが、集合$M=\left\{x \mid x\notin x\right\}$は次の矛盾を含む（Russell の逆理)。

$$M\in M\leftrightarrow M\in\left\{x \mid x\notin x\right\}\leftrightarrow M\notin M.$$

そこで、Zermelo は抽出公理の代りに次の分離(分出)公理を考案した。

$$\forall M\exists A\forall x\left(x\in A\leftrightarrow x\in M\wedge F\left(x\right)\right).$$

すなわち、扱う集合を任意の集合(M)の部分集合に限定するという着想である。

例えば、$M=\left\{\{0\},\{1\}\right\}$、$F\left(x\right): x\notin x$に分離公理を適用すると、Russell の逆理の式パターンは$M\in M\leftrightarrow M\in M\wedge M\notin M$となる。両辺とも偽であり全体として真と

なるから逆理は回避される(井関,1973)。こうすると、扱う集合は狭くなるが、安心して集合を使ってよいことになる。

また、集合の De Morgan の定理は命題論理との関連が重要である。p,q を命題とするとき

$$\neg(p \wedge q) \leftrightarrow (\neg p) \vee (\neg q)$$
$$\neg(p \vee q) \leftrightarrow (\neg p) \wedge (\neg q)$$
$$(p \Rightarrow q) \leftrightarrow \neg p \vee q$$

が成り立つ。命題の論理的等価についてはは4・2で考察する。

2・1・2 命題と集合

ここでは、方程式・不等式の解集合、命題の真と偽、条件、仮定・結論、$p \Rightarrow q$、反例、必要条件、十分条件、$p \Leftrightarrow q$ 、必要十分条件、同値、「p かつ q 」、「p または q」、「p でない」などの用語、論理に関する De Morgan の法則を扱う。

高校数学では述語 (predicate) という用語の代りに「条件」を使う。述語を性質または条件ということがある（前原,1966）し、必要条件・十分条件という用語と整合性がある。ただ、数理論理学では条件より述語が一般的な用語なので、以下述語を用いる。解集合は等号・不等号で表された述語を真にする集合であり、方程式や不等式の解や図形と方程式・不等式の考察に有効である。

このため、論理記号「∧かつ」、「∨または」、「¬否定」、「⇒ならば」、「⇔同値」のそれぞれと集合概念の共通部分、和集合、補集合との対応関係の理解が重要である。また、

$P=\{x \mid p(x)\}, Q=\{x \mid q(x)\}$ のとき、

$p(x) \Rightarrow q(x)$ と $P \subset Q$ との対応関係、

$p(x) \Leftrightarrow q(x)$ と $P=Q$ との対応関係の理解も重要である。これらについては、**4**、**5** で体系的に考察する。

2・1・3　逆・裏・対偶

逆・裏・対偶、背理法について学ぶ。逆・裏・対偶では命題の真偽値がどうなるかが重要である。命題の真偽値は真T、偽Fの値割り付け（valuation）表（真理値表）による。２つの命題の同値関係は真偽値の一致・不一致を調べる。これは命題の解釈の問題で4・2で考察する。述語の場合の解釈は難度が高くなる。$p(x) \Rightarrow q(x)$ の真偽は2・1・2で述べた集合の包含関係の解釈で証明し、8・2の例8で考察する。

２－２「数学A」

「図形の性質」の平面図形では、三角形や円に関する基本的な性質について、それらが成り立つことを証明できるようにする。

2・2・1　三角形の性質

直線と角、三角形の辺と角の大小、重心・内心・外心、Menelaos・Ceva の定理、Euler 線などを扱う。

2・2・2　円の性質

円周角の定理の逆、外接・内接四角形、接弦角、方べきの定理、２円の位置関係を扱う。図形の性質の演繹的構成は、Euclid の幾何学原論の方法論を踏襲するものである。その手段として証明という操作が使われていることの理解が主眼となる。

図形の証明は、公理Γ（または先行する定理）を基に、前提Cから三段論法という推論規則を用いて、幾何学的直観に導

かれつつ正しい命題A_iを積み上げてゆき結論Dを得るものである。記号で書けば$\Gamma, C, A_1, A_2, \cdots, D$あるいは$\Gamma, C \vdash D$となる。

もし、図形の性質を公理として記号化し、定理も記号化すれば、その証明は記号の世界だけの形式的演繹操作に帰するであろう。形式的証明については**5**で考察し、幾何の問題の導出原理に基づく証明例を８-２の例１５、**9**の例１６で与える。

２－３　「数学Ⅱ」

「いろいろな式」の「式と証明」で等式や不等式が成り立つことを証明できるようにする。

2･3･1 式と証明

ここでは、恒等式、係数比較法・数値代入法について学ぶ。係数比較法は恒等式を文字列と見做して対比する記号処理である。数値代入法は、変数に数値を代入して係数を求めるから記号列への値割り付け(assignment)として式を意味の世界に変換し解釈して解く方法である。一方、記号による代入(substitution)や置き換え(replacement)は、記号の世界から記号の世界に変換する操作である。

2･3･2 等式の証明

ここでは、等式、条件付きの等式の証明、比の値、比例式、連比を扱う。文字式の証明は記号操作で式の同値変形を行うもので純粋に記号の世界の中での処理である。例えば、記号処理ソフト mathematica を使って機械的に証明が出来る。

2･3･3 不等式の証明

ここでは、大小の判定、実数aは$a^2 \geq 0$、相加・相乗平均、平方の大小、絶対値を含む不等式を学ぶ。不等式の証明も記号操作の世界で出来る。

２－４「数学B」

「漸化式と数学的帰納法」で数列の漸化式による表現や数学的帰納法を扱う。漸化式と数列では、漸化式で表された数列の一般項を求める。また、数学的帰納法では、これを理解し、それを用いて簡単な命題を証明するとともに、事象の考察に活用する。

2･4･1 漸化式

ここでは、$a_1 = a$、$a_{n+1} = a_n + d$、$a_{n+1} = ra_n$、$a_{n+1} = pa_n + q$ などの漸化式を扱う。最後の式を以下で考察する。

n'は$n+1$(nの successor)、χは漸化式から決まる数論的関数(変域、値域が自然数)とし、$a_k = \varphi(k)$とおく。数列$\{a_n\}$は

$$a_{k+1} = \varphi(k') \equiv \chi(k, \varphi(k)) = p\varphi(k) + q$$

と原始帰納法(**3**で定義する)で定義できる。

$$\begin{aligned}
a_0 &= \varphi(0) = a. \\
a_1 &= \varphi(1) = \chi(0, \varphi(0)) \\
&= p\varphi(0) + q = ap + q. \\
a_2 &= \varphi(2) = \chi(1, \varphi(1)) \\
&= p\varphi(1) + q = ap^2 + pq + q.
\end{aligned}$$

・・・・・

よって、この数列を表す数論的関数χは原始帰納的関数のクラスに属する。

原始帰納的関数は一般帰納的関数である。また、一般帰納的関数はTuring計算可能であり、この逆も成り立つ。これが数列の漸化式をプログラムで記述し、計算できる根拠である。これについては、**3**で考察する。

2－4－2　数学的帰納法

ここでは、等式の証明、不等式の証明、整数の性質の証明を扱う。論理的推論には、普遍的な命題から特殊な命題を導く演繹とその逆を行う帰納がある。数学的帰納法は演繹法の一種であり、自然数nについての陳述$P(n)$を述語とみなして証明する方法である。数学的帰納法は次のように述べられる。

(1) $P(0)$が成り立つ

(2) $P(n) \Rightarrow P(n')$が成り立つ

このとき全てのn に対して$P(n)$が成り立つ。(1)は base of induction、(2)は induction step という。これは自然数の帰納的定義に依拠した証明方法である。

ちなみに、自然数の帰納的定義は

(1)　0は自然数

(2)　nが自然数ならば、n'は自然数

(3)　(1)(2)で与えられたもののみが自然数である

からなる。

Peano の自然数の公理系(Kleene, 1967)の第５番目にはこの数学的帰納法の原理が挙げられている。

以上､数理論理学に関連した教材内容を概観してきた。「ギリシャ以来、数学とはすなわち証明である」(N. Bourbaki, 1968) という有名な言葉があるが､その証明を支えている論理は、数学の屋台骨を成している。

3　漸化式と帰納的関数

2で述べた漸化式で定義される数論的関数は数列 $a_k\left(k=0,1,\cdots\right)$を$k$についての関数とみなすとき、原始帰納的関数である。一般的にこの関数φのクラスは次のように述べられる(Kleene, 1967)。

χ,ψを数論的関数とするとき、

(1) successor 関数　$\varphi(x)=x'.$

(2) 定数関数　$\varphi(x_1,\cdots,x_n)=q.$

(3) identity 関数　$\varphi(x_1,\cdots x_n)=x_i.$

(4) 合成関数

$$\varphi(x_1,\cdots x_n)$$
$$=\psi\left(\chi_1(x_1,\cdots x_n),\cdots,\chi_m(x_1,\cdots x_n)\right).$$

(5a) 1変数の原始帰納法

$$\varphi(0)=q,\varphi(y')=\chi\left(y,\varphi(y)\right).$$

(5b)一般の原始帰納法

$$\varphi(0,x_2,\cdots x_n)=\psi(x_2,\cdots x_n).$$
$$\varphi(y',x_2,\cdots x_n)$$
$$=\chi\left(y,\varphi(y,x_2,\cdots x_n),x_2,\cdots x_n\right).$$

(1)～(5b)を用いて作られる関数を原始帰納的 (primitive recursive) 関数という。
(5)で作られる関数はただ一つ存在する。

例１　次の漸化式を考える。

$$a_1=3,a_{n+1}=3a_n+2^n.$$

$\varphi(1)=3,\chi\left(n,\varphi(n)\right)=3\varphi(n)+2^n$とおけば(5a)の形になり原始帰納的となる。

階乗　$n!$も次のように原始帰納的となる。

$$\varphi(0)=1.$$
$$\varphi(n')=\chi(n,\varphi(n))=(n+1)\varphi(n).$$

関数$a+b, a\cdot b, a^b, a!, |a-b|, \cdots$も原始帰納的である。

また、集合Sの特徴関数C_sを

$C_S(x)=0(x\in S), 1(x\notin S)$と定義すると

これが原始帰納的であればSは帰納的集合であること、また述語pはその真理集合

$\{x \mid p(x)\}$が原始帰納的ならば原始帰納的

述語であることが定義できる。こうして、原始帰納的関数の概念を関数から集合や述語へ拡張していく。さらに、原始帰納的関数にμ作用素(等式を満たす最小のyの値が関数値)を施して作られる関数hを定義できる。

$$h(y,x_1,\cdots x_n)=\mu y(f(y,x_1,\cdots x_n)=0).$$

これらの関数のクラスを一般帰納的関数(general recursive)という。原始帰納的関数は一般帰納的関数である。この逆は成り立たない(反例; Ackermann関数)から一般帰納的関数のクラスの方が大きくなる。

なお、一般帰納的関数のクラスはTuring計算可能な関数のクラスと一致することが分かっている(Davis, 1958)。

4　命題論理と述語論理

命題と述語の論理体系を考察する。

4－1　自由変数と束縛変数

“変数とは・・”という数学の定義はない(井関, 1973)。“独立変数”などとその冠が与えられて定義されているに過ぎない。数学で扱う数式には“変数”が用いられ、例えば、教科書では次のような式の変数記号が出てくる。

$$2x^2-x-1=0, \log_2 x\leq -1. \quad \cdots ①$$

$$(x+y)^2=x^2+2xy+y^2. \quad \cdots ②$$

$$\sin(x+y)=\sin x\cos y+\cos x\sin y. \quad \cdots ③$$

$$\lim_{x\to 0}\frac{\sin x}{x}=1. \quad \cdots ④$$

$$\int_0^t xdx=\frac{1}{3}t^3. \quad \cdots ⑤$$

$$x=a(t-\sin t), y=a(1-\cos t). \quad \cdots ⑥$$

①は方程式、不等式でありxは特定の値(範囲)を満たす未知数を表す変数である。②、③は恒等式で、x, yは任意の数を表す変数である。④の初めの式のxは0に収束する変数を表す。2番目の式のxは任意の値を表す変数である。⑤のtは積分変数と称され積分操作を表す変数である。⑥のtはパラメータと称され、解析幾何特有の任意の値を表す変数である。

④、⑤のxは式が表す意味にかかわる変数である。一方、他の式の変数は量や命題の真偽にかかわる。数理論理学においては、前者を束縛変数, 後者を自由変数とタイプ分けしている。形式的には、論理式で表すとき全称作用素$\forall x$、存在作用素$\exists x$の有効範囲(scope)内の変数xを束縛変数といい、それ以外は自由変数という。

未知数として出てくる①の変数は、それにある定数の代入で、＝の意味(真・偽)が決まるので自由変数である。④のx　、⑤のtは勝手な値の代入で無意味になるから束縛変数である。ところが難しいのは②, ③である。

恒等式は勝手な値割り付けで常に成り立つので自由変数だが、この式を

$$\forall x \forall y \left((x+y)^2 = x^2 + 2xy + y^2 \right).$$

と論理式で表すと x, y は束縛変数になる。形式化されていない数学で扱う文字 x がどういうタイプの変数であるかはその式の表現で判断する必要がある。

4・2　命題論理

命題論理は命題の内容に立ち入らずにその真偽（真理値）のみを問題にする。命題論理の論理式を定義する。

(1) 命題定数 T, F 、命題変数は論理式である(原子命題)

(2) φ, ψ が論理式ならば、

$\neg\varphi$ 、$\varphi \wedge \psi$ 、$\varphi \vee \psi$ 、$\varphi \supset \psi$ 、$\varphi \leftrightarrow \psi$ は論理式である

(3) (1),(2)で出来たもののみが論理式である

命題論理式は、命題変数 $X_1, \cdots, X_n$ が論理記号の $\neg$(否定)、$\wedge$(連言)、$\vee$(選言)、$\supset$(含意)、$\leftrightarrow$(同値)を用いて構成される。

このように生成された論理式はその形が異なっていても同じ真理値をとるものは特に論理的に同値といい、$\varphi \equiv \psi$ と記す。この同値変形には以下の関係が用いられる。

可換律　$\varphi \wedge \psi \equiv \psi \wedge \varphi, \varphi \vee \psi \equiv \psi \vee \varphi.$

結合律
$$(\varphi \wedge \psi) \wedge \chi \equiv \varphi \wedge (\psi \wedge \chi).$$
$$(\varphi \vee \psi) \vee \chi \equiv \varphi \vee (\psi \vee \chi).$$

分配律
$$(\varphi \wedge \psi) \vee \chi \equiv (\varphi \vee \chi) \wedge (\psi \vee \chi).$$
$$(\varphi \vee \psi) \wedge \chi \equiv (\varphi \wedge \chi) \vee (\psi \wedge \chi).$$

吸収律

$$\varphi \wedge (\varphi \vee \psi) \equiv \varphi. \quad \varphi \vee (\varphi \wedge \psi) \equiv \varphi.$$

$\supset$ の別表現　$\varphi \supset \psi \equiv \neg\varphi \vee \psi.$

$\Leftrightarrow$ の別表現　$(\varphi \supset \psi) \wedge (\psi \supset \varphi).$

二重否定の除去　$\neg\neg\varphi \equiv \varphi.$

De Morgan の法則

$$\neg(\varphi \vee \psi) \equiv \neg\varphi \wedge \neg\psi.$$
$$\neg(\varphi \wedge \psi) \equiv \neg\varphi \vee \neg\psi.$$

対偶の法則　$\varphi \supset \psi \equiv \neg\psi \supset \neg\varphi.$

6・1で、これらが命題変数への真理値割り当てによる解釈で成り立つことが分かる。

$f(x_1, \cdots, x_n) : \{0,1\}^n \to \{0,1\}$ を Boole 関数と呼べば、論理式は Boole 関数と考えられる。任意の Boole 関数は標準形に同値変形で変換できる。すなわち、$\wedge, \vee, \neg$ の組み合わせのみで表現できることが分かっている。これは論理回路の設計に応用される。

なお、最小限の論理素子を使うためには次の $NAND, NOR$ 素子を用いる。

$\neg(x \wedge y)$ が $NAND$　,$\neg(x \vee y)$ が NOR

すべての論理式はこのいずれかで表現できる。その根拠は

$$\neg x \equiv \neg(x \wedge x) \equiv \neg(x \vee x); NAND, NOR$$
$$x \wedge y \equiv \neg(\neg x \vee \neg y); NOR$$
$$\equiv \neg(\neg(x \vee x) \vee \neg(y \vee y)); NOR$$
$$x \vee y \equiv \neg(\neg(x \wedge x) \wedge \neg(y \wedge y)); NAND$$

が成り立つからである。

4－3　述語論理

述語論理は変数を含み内容に様々な対象を表す記号が入る。そこで、まず使う“記号”の定義から始める。

・特定の対象を表す定数記号
個体定数といい対象に依拠する

例２ $0,1,2,\cdots$, keiko, hide, yuito

・自由変数；$a,b,c,\cdots$
・束縛変数；$x,y,z,\cdots$
・関数記号；$f,g,f_1^1,f_2^1,\cdots,f_m^n,\cdots$

例３ $f_1^2;plus,f_2^2;times$

・述語記号；$p,p_1^1,p_2^1,\cdots,p_m^n$
考察対象に依拠する

例４ $p_1^2;*_1=*_2$, $p_2^2;*_1<*_2$, $\cdots$

・論理記号；$\neg$, $\wedge$, $\vee$, $\supset$, $\forall$, $\exists$

次に、項(term)を定義する。
(1) 特定の対象を表す記号（定数)、変数は項である
(2) $f^n(*_1,*_2,\cdots,*_n)$ はn変数の関数とし、$t_1,\cdots,t_n$ が項ならば、$f^n(t_1,\cdots,t_n)$ は項である
(3)(1),(2)で定義されたもののみが項である

項の集合Tを対象領域という。ある集合の内包的定義$M=\{a\,|\,A(a)\}$において、$A(a)$ は自由変数aを含む文である。

次に、このような文を表現する“論理式(formula)”を定義する。
(1) $p^n(*_1,\cdots,*_n)$ はn変数の述語、$t_1,\cdots,t_n$ が項ならば、$p^n(t_1,\cdots,t_n)$は論理式（素論理式）である
(2) α,βが論理式のとき、$\neg\alpha,\alpha\wedge\beta,\alpha\vee\beta,\alpha\supset\beta$はすべて論理式である
(3) 論理式$\alpha(a)$において、自由変数aをこの論理式に含まれない変数xで置き換えたものの前に$\forall x,\exists x$を付けてできる$\forall x\alpha(x),\exists x\alpha(x)$は論理式である
(4)(1),(2),(3)で構成されたもののみが論理式である

(1)～(3)で生成される論理式を整合論理式(well formed formula)という。

命題論理で考察したように述語論理でも同値変形が考えられる。２つの論理式φ,ψが論理的に同値であるとは、すべての解釈、すべての変数割り当てで同じ真理値を持つ場合とし、$\varphi\equiv\psi$と記す。命題論理で成り立つものはそのまま成り立ち、これに以下のものを加える。
φがyを自由変数として含まないとき

$$\forall x\varphi(x)\equiv\forall y\varphi(y),\exists x\varphi(x)\equiv\exists y\varphi(y).$$

$$\neg\forall x\varphi\equiv\exists x\neg\varphi,\neg\exists x\varphi\equiv\forall x\neg\varphi.$$
$$(\forall x\varphi)\wedge(\forall x\psi)\equiv\forall x(\varphi\wedge\psi).$$

$$\left(\exists x\varphi\right)\vee\left(\exists x\psi\right)\equiv\exists x(\varphi\vee\psi).$$

φ が x を自由変数として含まないとき

$$\varphi\wedge\left(\forall x\psi\right)\equiv\forall x\left(\varphi\wedge\psi\right).$$
$$\varphi\wedge\left(\exists x\psi\right)\equiv\exists x\left(\varphi\wedge\psi\right).$$
$$\varphi\vee\left(\forall x\psi\right)\equiv\forall x\left(\varphi\vee\psi\right).$$
$$\varphi\vee\left(\exists x\psi\right)\equiv\exists x\left(\varphi\vee\psi\right).$$

以上の式変形を用いれば任意の論理式は冠頭(前置)標準形(prenex normal form)に変形できる。その形は $Q_1x_1\cdots Q_mx_m\varphi$ である。

Q_i は量化子 $\forall,\exists$ のいずれかである。命題論理で述べたように φ は $\wedge,\vee,\neg$ を用いて表せるが、さらに連言標準形(CNF)または選言標準形に同値変形できることが分かっている。

また、Q_i のうち $\exists$ 記号を除去するため、定数記号(Skolem 定数)と関数記号(Skolem 関数)を導入すれば Skolem 標準形とよばれる形に変形できる。結果として任意の論理式は次の形に変形されることが分かる。

$\forall x_1\cdots\forall x_m\left(C_1\wedge\cdots C_n\right).$

C_i は節(clause)といい、素論理式の選言 $C_i=l_{i1}\vee\cdots\vee l_{im}$ である。l_{ij} は p または $\neg p$ でリテラル(literal)という。この２つは相補対(complementary pair)という。

例５　$\exists x\forall y\forall z\exists wp\left(x,y,z,w\right).$

束縛変数 x にはある定数記号 a を代入、変数 w には変数 y,z の関数 $f\left(y,z\right)$ を代入すれば、$\forall y\forall zp\left(a,y,z,f\left(y,z\right)\right)$ となる。

７，８においては、述語論理式を考察する際、これを Skolem 標準形に同値変形しこの節集合 $\left\{C_1,C_2,\cdots C_n\right\}$ を対象にする。

５　形式的体系と証明可能性

４で定義した論理式を考察するに当たっては、構文論(syntax)の立場と意味論(semantics)の立場がある。構文論では、公理と推論規則を定めた形式的体系(formal system)の中で、記号操作のみで演繹的に論理式が証明可能(provable)か否かを考察する。他方、意味論では論理式に対する解釈を基に、それが恒真(tautology)であるか否かを調べる。

ある論理式が証明可能ならば恒真であることを健全性といい、この逆が成りたつことを完全性という。命題計算では、ある論理式が証明可能であることと恒真であることとは同値となる。また、述語計算でも、証明可能な論理式はあらゆる解釈において真(妥当 validity)なることがいえ、この逆も成り立つ。これが成り立つことを保証するのが完全性定理(*Gödel*, 1930)である。結局、構文論でも意味論でもどちらで考察を進めても構わないことになる。以下、形式的体系と証明可能性について考察する。

５・１　Hilbert の体系

形式的な論理体系には次のような Hilbert の体系 H (Kleene,1967)がある。

H1a　$A\supset\left(B\supset A\right)$　　H2　$\dfrac{A\quad A\supset B}{B}$

H1b $\left(A\supset B\right)\supset\left(A\supset\left(B\supset C\right)\supset\left(A\supset C\right)\right)$

H3　$A\supset\left(B\supset A\wedge B\right)$

H4a　$A \wedge B \supset A$　H4b　$A \wedge B \supset B$

H5a　$A \supset A \vee B$　H5b　$B \supset A \vee B$

H6　$(A \supset C) \supset ((B \supset C) \supset (A \vee B \supset C))$

H7　$(A \supset B) \supset ((A \supset \neg B) \supset \neg A)$

H8　$\neg\neg A \supset A.$　H9　$\dfrac{C \supset A(x)}{C \supset \forall x A(x)}$

H10　$\forall x A(x) \supset A(t)$　H11　$A(t) \supset \exists x A(x)$

H12　$\dfrac{A(x) \supset C}{\exists A(x) \supset C}$

(H9,H12 で C は自由変数 x を含まず、

H10,H11 の t　は $A(x)$ の x で自由変数)

命題論理の公理系 H1～H8 に H9～H12 を加えて述語論理の公理系とする。H2 は推論規則 Modus Ponens(MP、二段論法) という。

なお、H1~H12 を縮減した公理と推論規則として次の Lukasiewicz の公理系がある。この公理系を Γ とおく。

A1　$A \supset (B \supset A)$

A2　$(A \supset (B \supset C)) \supset ((A \supset B) \supset (A \supset C))$

A3　$(\neg B \supset \neg A) \supset (A \supset B)$

A4　$\forall x A(X) \supset A(t)$　t　は任意の項

A5　$\forall x (A \supset B(x)) \supset (A \supset \forall x B(x))$

A は x を自由変数として含まない

A6　MP (modus ponens)　$\dfrac{A \quad A \supset B}{B}$

A7　一般化　$\dfrac{A(X)}{\forall x A(x)}$

ただし、x は $A(x)$ を導く推論の前提に含まれない自由変数とする。

形式的体系の証明を以下の例でみる。

例 6

$$G = \begin{cases} man(Fermat) \\ \forall x(man(x) \supset fallible(x)) \end{cases}.$$

このとき $\Gamma, G \vdash fallible(Fermat)$ の証明は以下の 1. ~ 4.となる。

1.　$\forall x(man(x) \supset fallible(x))$　仮定

2.　$man(Fermat) \supset fallible(Fermat)$　A4

3.　$man(Fermat)$　仮定

4.　$fallible(Fermat)$　2. ,3. , A6(MP)

1.～4.が証明の過程であり、4.は定理となる。

一般に、公理系 H から導かれた論理式 φ は定理と呼ばれ $H \vdash \varphi$ と記す。このとき、φ は H から証明可能という。

5･2　形式的証明

以下のような演繹定理など多くの形式的な演繹によって多くの定理が導かれる。

[演繹定理]　$H, \varphi \vdash \psi$ ならば $H \vdash \varphi \supset \psi$

G が閉論理式の集合で φ が閉論理式とす

るとき、$G\cup\{\neg\varphi\}$ が矛盾するならば、$G\vdash\varphi$ が成り立つ。この証明は

$G'=G\cup\{\neg\varphi\}$ が矛盾するから、ある論理式 ψ について $G,\neg\varphi\vdash\psi$ と $G,\neg\varphi\vdash\neg\psi$ の両方が成り立つ。演繹定理により、

$G\vdash\neg\varphi\supset\neg\psi$ ・・・①

$G\vdash\neg\varphi\supset\psi$ ・・・②

H7 から

$(\neg\varphi\supset\neg\psi)\supset((\neg\varphi\supset\psi)\supset\varphi)$ ・・③

MP を①と③、その結果と②に２回使うと $G\vdash\varphi$ を得る。

これは、背理法の原理である。その他にも多くの論理の定理があるが割愛する。

６　論理式の解釈

意味論の立場で論理式を考察する。文字の抽象性は意味付けが多義であることによる。この意味付けとは解釈に他ならない。

例えば、コンピュータ・プログラムはプログラム言語の文法に則って coding された記号列に過ぎないが、コンパイラーによってある動作を行う機械語に翻訳（解釈）されコンピュータを動かす binary program となる。コンピュータは多種類あるから一つの言語のコンパイラーもその数だけあるように解釈は無数にある。

以下、論理式の解釈について考察する。

６‐１　命題論理式の解釈

具体的な個体領域 $D=\{T,F\}$ を考える。D の要素を真理値という。また、論理式 p に含まれるすべての命題変数 $X_i\ (1\le i\le n)$ に真理値を割り当てるとき、それを真理値割り当て(truth assignment)という。この割り当てを論理式の解釈という。命題論理式の解釈に対しては、次式を満たす写像 I となる。

G：命題論理式の集合、$G\ni\varphi\to I(\varphi)\in D.$

(1) $I(T)=T, I(F)=F.$

(2) X_i は命題変数で、t_i を真理値割り当ての値とするとき $I(X_i)=t_i.$

(3) 論理記号の解釈

$I(\neg\varphi)=\neg I(\varphi), I(\varphi\wedge\psi)=I(\varphi)\wedge I(\psi).$

$I(\varphi\vee\psi)=I(\varphi)\vee I(\psi).$

$I(\varphi\supset\psi)=I(\varphi)\supset I(\psi).$

$I(\varphi\leftrightarrow\psi)=I(\varphi)\leftrightarrow I(\psi).$

(1)～(3)によって、任意の命題論理式に D の要素が割り当てられる。４‐２で挙げた同値変形が確かめられる。

命題論理式 φ が、真理値割り当てに対して $I[\varphi]=T$ のとき恒真式(tautology)といい、$\models\varphi$ と記す。恒真式は絶対的な真実を表す。

また、ある条件下で成り立つという場合には、限定的な真理値割り当て（解釈）I' で $I'(\varphi)=T$ となる場合であり、I' は φ を充足する、I' は論理式 φ のモデルという。

また、φ は充足可能(satisfiable) という。また、論理式の集合 Γ と論理式 φ があるとき、Γ を充足する真理値割り当てが φ も充足するとき $\Gamma\models\varphi$ と記す。

６・２　述語論理式の解釈

４・３で定義した整合論理式への解釈を考える。Dを具体的な個体領域（解釈空間）とする。個体領域の例は自然数、実数、特定のモノの集合が考えられる。ある論理式に対する解釈(D,I) を以下のような写像I_σにより定義する。変数がないときはIを使う。

(1) 個体定数cには$I(c)\in D$を、変数記号にはD上を動く変数を対応させる

(2) 関数記号f^n に対しては、関数 $I(f^n);D^n \to D$を対応させる

(3) 述語記号p^nに対しては命題関数 $I(p^n);D^n \to \{T,F\}$を対応させる

σは、述語に含まれる自由変数への一つの値割り当て(assignment)である。

このとき、(1)～(3)より次式が成り立つ。

$I_\sigma(c)=I(c).$,　$I_\sigma(x)=\sigma(x).$

$I_\sigma(f(t_1,\cdots,t_n))=I(f)(I_\sigma(t_1),\cdots,I_\sigma(t_n)).$

$I(p(t_1,\cdots,t_n))=I(p)(I_\sigma(t_1),\cdots,I_\sigma(t_n)).$

$I_\sigma(\neg\alpha)=\neg I_\sigma(\alpha).$

$I_\sigma(\alpha\wedge\beta)=I_\sigma(\alpha)\wedge I_\sigma(\beta).$

$I_\sigma(\forall x\alpha)$は$D$の任意の要素$a$ に対して

$I_{\sigma\{x:=a\}}(\alpha)=T$とする。

$I_\sigma(\exists x\alpha)$は$D$のある要素$a$について

$I_{\sigma\{x:=a\}}(\alpha)=T$とする。

与えられた述語論理式が自由変数を含まないとき閉論理式という。この場合の解釈は命題論理と同様に一意的に決まる。

自由変数を含む場合、自由変数への値割り当てσが決まれば解釈(D,I_σ)が決まる。よって、述語論理式の解釈は無数にある。

解釈(D,I_σ) において$I_\sigma(\alpha)=T$のとき、$(D,I)\models_\sigma \alpha$と記し、$\alpha$は妥当(valid)という。

特に、αが閉論理式のとき解釈(D,I) はαを充足するという。閉論理式の集合Gのすべての論理式が充足されるとき(D,I) はGを充足するといい、$(D,I)\models G$と記す。

例７　個体領域　$D=\{0,1,2,\cdots\}$

・個体定数；$zero, one, two,\cdots$

$I(zero)=0, I(one)=1,\cdots.$

・関数記号；add,times

$I(add)=*_1+*_2.$,　$I(times)=*_1\cdot *_2.$

・述語記号；eq　　$I(eq)=equal(*_1,*_2).$

$equal$ は次の命題関数であり

$equal(x,y)=\begin{cases}T\ (x=y)\\ F\ (x\neq y)\end{cases}$ とする。

$$
\begin{aligned}
&I\left(add\left(zero,one\right)\right)\\
&=I\left(add\right)\left(I\left(zero\right),I\left(one\right)\right)\\
&=I\left(zero\right)+I\left(one\right)=0+1=1.
\end{aligned}
$$

$$
\begin{aligned}
&I\left(eq\left(one,one\right)\right)\\
&=I\left(eq\right)\left(I\left(one\right),I\left(one\right)\right)\\
&=equal(1,1)=T.
\end{aligned}
$$

例８　2-1-2 で述べた論理式 $p(x)\Rightarrow q(x)$ を再考察する。この式は同値変形を施すと

$\forall x\left(\neg p(x)\vee q(x)\right)$ となる。

解釈は、個体領域 D を実数の集合とし、

$D\ni\forall a$ に対して $\sigma(x)=a$ とする。

$$
\begin{aligned}
P&=\left\{x\mid I_\sigma\left(p\left(x\right)\right)=T\right\}\\
Q&=\left\{x\mid I_\sigma\left(q\left(x\right)\right)\}=T\right\}
\end{aligned}
$$

とおくと

$I\left(p\left(a\right),a\in P\right)=T$, $I\left(q\left(a\right),a\in Q\right)=T$

である。このとき、$P\subset Q$ が成り立つことと

$I_\sigma[\neg p(x)\vee q(x)]$ が $\forall a\in D$ に対して恒真式となることは論理的に同値である。その証明を以下で述べる。

　$P\subset Q$ を仮定する。まず次式が成り立つ。

$$
\begin{aligned}
&I_\sigma\left(\neg p(x)\vee q(x)\right)=\neg I_\sigma\left(p(x)\right)\vee I_\sigma\left(q(x)\right)\\
&=\neg I\left(p(a)\right)\vee I\left(q(a)\right).
\end{aligned}
$$

$a\in P$　ならば $a\in Q$　よって $I\left(q\left(a\right)\right)=T.$

$a\notin P$　ならば $\neg I\left(p\left(a\right)\right)=T$ となる。

いずれにせよ $I\left(\neg p\left(a\right)\vee q\left(a\right)\right)=T$ がいえる。逆も明らかである。

　5 で考察した形式的体系で演繹される証明可能性と **6** で考察した解釈（モデル）に基づく充足可能性の関連はどうなるのだろうか。いま、G という閉論理式の集合とし、$G\ni\varphi$ とする。

　もし、$G\vdash\varphi$ ならば $G\models\varphi$ が成り立つとき健全性(soundness)が成り立ち、逆が成り立つとき完全性(completeness) が成り立つという。命題論理と述語論理においては健全性と完全性は同値であることが分かっている（Gödel の完全性定理）。

7　Herbrand の定理

　この定理は決定問題の可解な場合を扱う基本的なツールであり、定理の自動証明の基礎になっている(Andrews,1986)。以下、定理で用いる概念構成を与える。

7・1　Herbrand 領域

　基礎項(ground term)とは、関数に含まれる変数に個体定数を代入したものである。いま、G は閉論理式の集合、F はそれに含まれる個体定数と関数の集合とする。

　G の Herbrand 領域(Herbrand universe) H とは、G に含まれる関数の変数に F の要素を代入してできる基礎項の集合である。関数記号を含むときは可附番無限集合になる。

　例９　閉論理式の集合を

$$
G=\left\{
\begin{array}{l}
is(chichi\left(yuito\right),hide),\\
is(haha\left(hide\right),keiko),\\
\forall x\forall y\left(is\left(chichi\left(x\right),y\right)\supset oya\left(y,x\right)\right),\\
\forall x\forall y\left(is\left(haha\left(x\right),y\right)\supset oya\left(y,x\right)\right)
\end{array}
\right\}
$$

　とする。G の個体定数は

$hide, yuito, keiko$ 、関数記号は $haha, chichi$ であるから、

$$F=\{hide, yuito, keiko, haha, chichi\}.$$

$$H=\left\{\begin{array}{l} yuito.hide, keiko, \\ haha(yuito), chichi(yuito), \\ \cdots, \\ chichi(haha(yuito)) \\ ,\cdots \end{array}\right\}.$$

7・2　Herbrand 解釈、モデル

閉論理式の集合 G に対して、述語に含まれる変数に H の要素を代入した命題論理式の全体を Herbrand 基底(Herbrand basis) B_G という。

例 10　例 9 において、述語は is, oya であるから Herbrand 基底は

$$B_G=\left\{\begin{array}{l} is(yuito, yuito), is(yuito, hide), \\ is(yuito, keiko), \cdots \\ is(chichi(yuito), yuito), \cdots \\ oya(yuito, yuito), \\ oya(yuito, hide), \cdots \end{array}\right\}$$

となる。

以上の概念を基に、閉論理式の集合 G の Herbrand 解釈 (H, I) を与える。

解釈空間を Herbrand 領域 H とし、個体定数 c には $I(c) \in H$ を 、関数 f には $I(f)(t_1, \cdots, t_m) \in H$ を、述語 p には次のように命題論理式をそれぞれ対応させる。

$$I(p)(t_1, \cdots t_n)=T \leftrightarrow p(t_1, \cdots t_n) \in B_G$$

このとき、次式を満たす素論理式の集合 M を Herbrand 解釈 (H, I) という。

$$M=\{p(t_1, \cdots, t_n) \mid I(p)(t_1, \cdots t_n)=T, t_i \in H\} \subset \mathrm{B}_G.$$

このとき、$(H, I) \models G$ が成り立ち、M は G の Herbrand モデルという。

無矛盾な閉論理式の集合は Herbrand モデルをもつことがいえる。

例 11　例 9 において、以下の集合 M は G の Herbrand モデルとなる。

$$M=\left\{\begin{array}{l} is(chichi(yuito), hide), \\ is(haha(hide), keiko) \\ oya(keiko, hide), oya(hide, yuito) \end{array}\right\}.$$

以下、整合論理式を SkolemCNF に変形した節の集合 $\{C_1, \cdots C_n\}$ を考察対象にする。

変数を含まないリテラルを基礎リテラル(ground literal)といい、この集合を基礎節(ground clause)という。

いま、$S=\{C_1, \cdots C_n\}$ とする。節 C_i に含まれるリテラルの変数に S の Herbrand 領域 H の基礎項を代入して得られる基礎節の集合を $H(S)$ と記す。$H(S)$ は Herbrand 基底に対応する概念である。相補対を含まない基礎節はモデルに対応する概念である。

モデル M が基礎節の集合 S のモデルとなるのは、S のどの基礎節 C も M の要素を含むときとする。このとき M は $H(S)$ のモデルとなる。モデルは複数存在する。

節の有限集合Sのモデルが存在するときSは充足可能(satisfiable)であると定義する。モデルがないときは充足不能という。空節(empty clause)（□と記す）を含むどんな節の集合も充足不能となる。

節の有限集合Sが充足可能であるための必要十分条件はある Herbrand 解釈がSを充足することである。

7・3　Herbrand の定理

この定理は次のように表現される。

「Sが節の有限集合、Hが Herbrand 領域であるとき、Sが充足不能であるための必要十分条件は次の1(または2)である。

1　$H(S)$のある有限部分集合が充足不能となることである

2　Hの有限部分集合Pを用いてできる$P(S)$が充足不能となることである」

2では$P_i \subseteq P_{i+1}, \cup_{i=1}^{\infty} P_i = H$が成りたつ。

$P_i(S)$の充足不能性を調べるアルゴリズムが考案されたが、効率上の困難があった(Gilmore&Davis のアルゴリズム)。

8　導出原理と反駁

7-3 のアルゴリズムの効率化の中で、導出原理が考案された(Robinson,1965)。

8・1　命題論理の導出原理

CとDを次のような基礎節とし、LとMは相補対とする。

$$C \supseteq L = \{p\}, D \supseteq M = \{\neg p\}.$$

$$R = (C - L) \cup (D - M)$$をCとDの基礎導出節(ground resolvent)という。

次に、基礎節の集合Sに対する基礎導出(ground resolution) $R(S)$を

$R(S) = S \cup$ {Sの要素同士のすべての対に対する基礎導出節} と定義する。この操作を導出原理 (Resolution Principle)という。n回この操作を行った結果できる集合を$R^n(S)$と記す。このとき次式が成り立つ。

$$S = R(S) \subseteq R^2(S) \subseteq \cdots \subseteq R^n(S) \subseteq \cdots.$$

節の個数は有限個なのでいずれかのnで$R^n(S) = R^{n+1}(S)$となり、生成される基礎導出節の個数は減少していく。空節□の導出を反駁(refutation)という。

[基礎導出定理]

「基礎節の有限集合Sが充足不能であるための必要十分条件は、あるnに対して$R^n(S) \ni \square$となる、すなわち反駁が存在することである」

例 12　$S = \{\neg p \vee q, p \vee q, \neg q\}.$

このとき次のような反駁が得られる。

1　$C_1 : \neg p \vee q, C_3 : \neg q \to R_1 : \neg p$

2　$C_2 : p \vee q, R_1 : \neg p \to R_2 : q$

3　$C_3 : \neg q, R_2 : q \to R_3 : \square$

別の反駁もある。

1　$C_1 : p \vee q, C_3 : \neg q \to R_1 : p$

2　$C_1 : \neg p \vee q, R_1 : p \to R_2 : q$

3 $C_3 : \neg q, R_2 : q \to R_3 : \square$

いずれにせよ、Sは充足不能となる。

8－2 述語論理の導出原理

変数v_i 、項t_iとするとき、v_iをt_iで置き換える代入を次のように表記する。

$$\theta = \{v_i / t_i\}, i = 1, 2, \cdots, n.$$

論理式をαとβ、代入をθとする。もし$\alpha\theta = \beta\theta$となるとき$\alpha, \beta$は単一化が可能といい、$\theta$は単一化代入(unifier)という。このうちで代入回数がより少ないもの、すなわち代入λ、θに$\lambda = \theta\xi$が成り立つ場合、θ、ξはλより一般的と考えて最汎単一化(most general unifier、略して mgu)という。mgu を用いると最小限の代入で 2 つの論理式を一致させられる。

例 13 $W = \{p(x, y), p(z, f(b))\}.$

$$\theta = \{x / z, y / f(b)\}.$$
$$\sigma = \{x / a, y / f(b), z / a\}.$$

このときθ, σは単一化代入となる。

$\sigma = \theta\{z / a\}$だからθが mgu になる。

単一化(unification)アルゴリズムが導出の過程で使われる。

そこで述語論理の導出を定義する。いま、C、Dを変数を共有しない(共有する場合は変数名を変更しておく)節とし、それぞれのリテラルを L、M、mgu をσとする。$L\sigma$と$M\sigma$が相補対になるとき、

$R = (C - L\sigma) \cup (D - M\sigma)$ をC, Dの導出節(resolvent)という。また、節の集合Sに対して、

$R(S) = S \cup$ {Sのすべての要素同士からなる導出節} を求める操作を 8-1 と同様に導出原理という。この導出を繰り返すと

$$S = R^0(S) \subseteq R(S) \subseteq \cdots \subseteq R^n(S) \subseteq \cdots$$

となる。

[導出定理]

「節の有限集合Sが充足不能であるための必要十分条件は、あるnに対して$R^n(S) \ni \square$が成り立つことである」

導出は、$R^n(S) \ni \square$を見つけるまで続けられることになる。但し、その停止性の保証はない。何故なら、Curch の定理により一般に一階の述語論理における決定手順は存在しないからである。

例 14 反駁の簡単な例

$S = \{p(x) \vee p(f(y)), \neg p(f(g(a)))\}.$

1 $p(x) \vee p(f(y)).$

2 $\neg p(f(g(a))).$

3 $p(x)$ 1, 2, $\{y / g(a)\}.$

4 $\square$ 2, 3, $\{x / f(g(a))\}.$

8－3 導出原理による証明例

$S \models \alpha$を証明するには、反駁しようとする

論理式αの否定Sに含めた集合$S\cup\{\neg\alpha\}$に対して導出の操作を繰り返し行うことになる。その結果、空節□が生成されれば$S\cup\{\neg\alpha\}$は反駁されるので充足不能となり、$S\models\alpha$が結論できる。以下で、証明の実例を考察する。

例 15　初等幾何の定理の証明(林, 1973)
定理
「二等辺三角形の両底角は相等しい」

証明は次の 3 ステップを経る。

Ⅰ　幾何学の知識を論理式化する。

辺、角の満たす条件を 2 つの述語を用いて表現する。点はx, y, z, u, v, w　で表す。

$$p(x,y,u,v);xy=uv.$$
$$q(x,y,z,u,v,w);\angle xyz=\angle uvw.$$

1 二角形の合同定理(二辺挟角相等)

$$\left.\begin{array}{l} xy = uv \\ xz = uw \\ \angle yxz = vuw \end{array}\right\}$$ ならば $\angle xyz = \angle uvw$　を

論理式で表すと次のようになる。

$$\forall x\forall y\forall z\forall u\forall v\forall w\begin{pmatrix} p(x,y,u,v)\wedge \\ p(x,z,u,w)\wedge \\ q(y,x,z,v,u,w) \\ \supset q(x,y,z,u,v,w) \end{pmatrix}.$$

2 辺の相等（対称律）

$xy = yx$　、$xy = uv$　ならば　$uv = xy$　を表す論理式は次のようになる。

$$\forall x\forall y\forall u\forall v\big(p(x,y,u,v)\supset p(u,v,x,y)\big).$$

3 角の相等

$$\angle yxz = \angle zxy$$
$$\forall x\forall y\forall z\big(q(y,x,z,z,x,y)\big).$$

4 仮定

$$AB = AC \qquad p(A,B,A,C).$$

5 証明する命題

$$\angle ABC = \angle ACB.$$
$$q(A,B,C,A,C,B).$$

Ⅱ　これらを節の集合に変形する。

$$C_1:\begin{Bmatrix} \neg p(x,y,u,v),\neg p(x,z,u,w) \\ \neg q(y,x,z,v,u,w),q(x,y,z,u,v,w) \end{Bmatrix}.$$
$$C_2:\{\neg p(x,y,u,v),p(u,v,x,y)\}.$$
$$C_3:\{q(y,x,z,z.x,y)\}.$$
$$C_4:\{p(A,B,A,C)\}.$$
$$C_5:\{\neg q(A,B,C,A,C,B)\}.$$

C_5は定理の結論 5 の否定である。

Ⅲ　$C_1 \sim C_5$　からの反駁を作成する。
$R_1 \sim R_5$；導出節、θ,σ,λ　；mgu とすると次の導出節の列を得る。

$$R_1:\neg q(B,A,C,C,A,B),\neg p(A,C,A,B),$$
$$\neg p(A,B,A,C).$$
$$C_1\theta;C_5.$$
$$\theta=\{x/A,y/B,z/C,u/A,v/C,w/B\}.$$
$$R_2:\neg q(B,A,C,C,B,A),\neg p(A,C,A,B).$$
$$C_4;R_1.$$
$$R_3:\neg p(A,C,A,B).$$
$$C_3\sigma;R_2 \quad ,\sigma=\{x/A,y/B,z/C\}$$

$R_4 : \neg p\left(A,B,A,C\right).$

$C_2\lambda; R_3, \lambda = \left\{x/A, y/B, v/C\right\}.$

$R_5 : \square \qquad C_4; R_4$

よって、$\left\{C_i\right\}, i = 1,2,3,4,5$ は充足不能であり、定理は背理法により証明された。

9　PROLOG とその活用例

PROLOG は 1972 年頃、Alain Colmeraur と Pilippe Roussel が考案した述語論理処理のための言語である。この言語の処理系は、述語論理式 S とある論理式 α に対して $S \models \alpha$ かどうかを反駁で証明する。

このため、PROLOG 処理系では SLD 導出 (Selective Liner resolution for Definit clause) という操作がパターンマッチング、バックトラック処理で行われる。

処理の効率化という観点から扱う節は Horn 節に限定している。推論規則の後件が 1 つからなる自然演繹 NK や直観主義論理体系 LJ に似ている。Horn 節は以下のような論理式の節 C である。

$$p_1 \wedge p_2 \wedge \cdots \wedge p_m \supset q$$
$$\equiv \neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_m \vee q.$$
$$C = \left\{\neg p_1, \cdots, \neg p_m, q\right\}.$$

この式の PROLOG 表現は次式になる。

$$q : - p_1, p_2, \cdots, p_m. \qquad \cdots ①$$

q は head、$p_1, p_2, \cdots, p_m$ は body という。

head を A 、body を Δ とすれば①は「Δ ならば A である」を意味し、規則という。

Δ が空ならば、$q.$ と記す。これは恒真を表す命題であり、「常に真」を意味する事実といい、公理や前提の表現に用いられる。

head が空のときは以下のように記す。

$?- p_1, p_2, \cdots, p_m.$

これは、$p_1, p_2, \cdots, p_m \supset$ もしくは $\neg p_1 \vee \neg p_2 \vee \cdots \vee \neg p_m$ である。目標であり、質問又は結論の論理式を表すのに用いる。

例 16　例 15 の問題を証明する。

```
2  p(X,Y,Y,X).
   p(U,V,X,Y):-p(X,Y,U,V).
3  q(Y,X,Z,Z,X,Y).
   q(X,Y,Z,Z,Y,X).
1  q(X,Y,Z,U,V,W):-p(X,Z,U,W)
      ,q(Y,X,Z,V,U,W),p(X,Y,U,V).
4  p(a,b,a,c).
5  q(a,b,c,a,c,b).
```

SWI-PROLOG を用いてこの順に入力、コンパイルする。計算は以下のようになる。

```
[ trace ] 1 ? – q(a,b,c,a,c,b).
  Call (6):  q(a,b,c,a,c,b) ? creep
  Call (7):  p(a,c,a,b) ? creep
  Call (8):  p(a,b,a,c) ? creep
  Exit (8):  p(a,b,a,c) ? creep
  Exit (7):  p(a,c,a,b) ? creep
  Call (7):  q(b,a,c,c,a,b) ? creep
  Exit (7):  q(b,a,c,c,a,b) ? creep
  Call (7):  p(a,b,a,c) ? creep
  Exit (7):  p(a,b,a,c) ? creep
  Exit (6):  q(a,b,c,a,c,b) ? creep
True .
```

PROLOG は、入力文の上から下へ、body は左から右へ depth-first search で実行する。1 は、head と body に述語 q を含み無限ループに入るので 2,3 の次に入力する工夫をした。PROLOG は健全だが完全ではない。

例 17　Monkey-Banana 問題

問 題

「いま、地点 a に猿がおり、そばに箱がある。地点 b に吊るされたバナナを猿は取れるか」

I　問題状況を表現する関数、述語

集合 S は場の状態の集合、集合 P は猿のいる位置の集合、 $M=\{T,F\}$ は真理値の集合とする。

$$pushbox(x,s);P\times S\to S.$$
$$climbbox(s);S\to S.$$
$$grasp(s);S\to S.$$
$$onbox(s);S\to M.$$
$$atbox(x,s);P\times S\to M.$$
$$hold\_banana(s);S\to M.$$

$pushbox(x,s)$ は状態 s で位置 x まで箱を移動した状態、 $climbox(s)$ は状態 s で箱に登った状態、 $grasp(s)$ は状態 s でバナナをつかんだ状態への変化を表す関数である。

また、 $onbox(s)$ は状態 s で猿が箱の上にいるか否かで T,F となる述語とする。 $atbox(x,s)$ は、 s で猿が x にいるか否かで T,F となる述語、 $hold\_banana(s)$ は s でバナナをつかむか否かで T,F となる述語とする。

II 問題の状況を述語、関数で記述する。

a を猿の初期位置、 b を最終位置とする。

1　$\forall x\forall s\left(\begin{array}{l}\neg onbox(s)\supset\\ atbox(x,pushbox(x,s))\end{array}\right).$

2　$\forall s(onbox(climbbox(s))).$

3　$\exists s\left(\begin{array}{l}onbox(s),atbox(b,s)\supset\\ hold\_banana(grasp(s))\end{array}\right).$

4　$\forall x\forall s\left(\begin{array}{l}atbox(x,s)\supset\\ atbox(x,climbbbox(s))\end{array}\right).$

5　$\neg(onbox(a)).$

6　$\exists s(hold\_banana(s)).$

III　述語 1～5 を PROLOG で記述する

```
1  atbox(X,pushbox(X,Y)) :- not(onbox(Y)).
2  onbox(climbbox(S)).
3  hold_banana(grasp(S)):- onbox(S),
                           atbox(b,S).
4  atbox(X,climbbox(S)) :- atbox(X,S).
5  not(onbox (a)).
```

6 は否定の形で反駁を導出するので

$\forall s(\neg hold\_banana(s))$ となる。

これを PROLOG では ?-hold_banana(S). と記述する。

IV　SWI-PROLOG による実行

結果は以下の通り、最終の状態 S の値を合成関数の形で返してくる。

```
?-hold_banana(S).
S=grasp(climbbox(pushbox(b,a))).
```

これがバナナをつかむ行動手順を表す合成関数である。この操作は Resolution プログラムの unification で形成される代入リストを逆に辿ることから抽出することができ

る(林,1973)が、PROLOGを使えばいとも簡単にできてしまう。

なお、論理式の節の集合から手順を合成する技術は“プログラムの自動合成”の初歩とも考えられる。

１０　おわりに

論理の世界は奥が深く、一見、難解のように見えるが、少しずつ辛抱して学べば計算機科学とのつながりなど面白い視界が開けてくる。したがって、生徒にとって興味深い領域の教材化が求められている。

本稿で紹介した PROLOG を用いて発展的な論理教材を開発し、高校生にその一端を分かりやすく理解させることができれば、論理への興味関心は一層高まるにちがいない。

そのためにも、この分野に対する知識の体系的な理解を深めるため、数理論理学の素養を習得し、Herbrand の定理や導出原理、PROLOGの考え方を会得しておくことは意義のあることである。本稿がその誘い水となれば幸いである。

参考文献

[1]B.Andrews：数理論理学とタイプ理論、(小川倫了訳 1986),丸善(株)

[2]井関清志(193)：記号論理学(述語論理)、槇書店

[3]Kleene(1967)：Introduction to Meta-Mathematics,NORTH-HOLLAND PUBLISHING CO.

[4]佐藤泰介(1981)：導出原理による定理証明、情報処理 vol.22,No.11

[5]N.Bourbaki(1966): 数学原論　集合論１,(前原他訳 1968),東京図書

[6]竹内外史・八杉満利子(1987)：証明論入門、共立出版

[7]高橋陽一郎他(2011)：詳説・数学Ｉ、数学Ⅱ、数学Ａ，数学Ｂ、啓林館

[8]Davis(1958):Computability and Unsolvability,McGraw-Hill Book Co.(渡辺,赤訳,計算の理論,岩波書店,1966)

[9] 林雄一郎(1973)：RESOLUTIONに基づく問題解決システムとその一応用としての手順の合成について,東芝技術報告 RM-12176

[10] 林雄一郎(2005):人工知能研究における推論,第55回数学教育実践研究会

[11] Bratko (1986):PROLOG Programming for Artificial Intelligence,Addison-Wesley Publishing company,PROLOGへの入門(安部憲広訳)、近代科学社

[12] 前原昭二(1961)：数理論理学序説、共立全書[13]

[13] J.A.Robinson(1965)：A Machine -Oriented Logic Based on the Resolution Principle,J.ACM,Vol.12,No.1