Boole 革命再論

上村義明

これは第7回数学史シンポジウムの"ライプニッツの夢について"の続編である。ライプニッツは普遍記号学を夢想して将来，日常生活の推論に於いても，これを用いて処理されるようになると期待したといわれる。
その方向への breakthrough の第一歩は Boole による boolean value の発見であろう。これは"Boole 革命"と呼ぶに相応しい事件に思えるのだが，どうも歴史はこれを過小評価して，不完全燃焼のまゝ，Frege の方へ流れて行ってしまった。
そこで歴史の踏み違いを正して，どの様な景色が展開する筈だったかを復元しようというのがこの試みである。
（この様な発想の自由はこの研究会の寛容さに依っている）

数式に計算結果の値を対応づけるのに相似て，等式や不等式に boole 値を対応づける発想は自然な成り行きに思えるがその結果もたらされるものは決して小さくない。
証明過程は boole 値を保存する変形の一つと見做せるから式の変形は可能な限り機械化される。推論の際に伴う自然な感情の流れが，抽象化されて，無味乾燥な機械的操作に取って変わられ，その操作に別の感情が付随する。
それは推論の進化のようなもので，その新たな感情が推論を導くようになる。とすれば，もっと徹底的にこの作業をやり抜く必要がありはしないか？
"A＝B"を感覚的にどうとらえるかが鍵になる。
これを A と B が等しいという事実の認識の表現とみるのと，A と B が等しければ true，等しくなければ false の値をとる代物（object）と見るのとの間には可成りの落差がある。

後者には事実の重みがないので軽薄に見えるが、その代りに事実などの異物を安易に数学の中に持ち込む事を許さない態度が光っている。（警察ですら限度を守って民事への介入を避ける）
これは次のようにも云えよう。

"＝を演算子と見做す立場"

もはや＝は特別扱いを受けない。当然＝の定義はA,Bが関数の場合にはDomainに依存する。
Domainのすべての点で等しい時にtrue、そうでなければfalseとなる。それをbracket［　］で表わすのは自然であろう。この＝operatorは代数として特別に良い性質をもつわけではない。それ故Boolean algebraも一般代数系の一つと見做されるときは代数系に埋没されてしまう。
ところがboolean valueに於いては＝は≡であり、この operatorは極めて面白い性質をもっており、それがBoolean valueを面白くしていると云って過言でない。
ところが多くの人は $A \equiv B$ を $(A \leftarrow B) \wedge (B \leftarrow A)$ の云い換え位にしか思っていない。そこが理解されなかった。
≡こそ中心的存在であるのに！
≡は＝の性質とassociative, symmetricな性質を合わせもつ変幻自在の代物なのだ。これを活用しない手はない。
であるのにBooleは過少評価され、歴史は曲りそこねてFregeの方向に向って行った。計算の機械化の大道を踏みはずして、quantifierなどの些末にのめり込んで行く。必要がないと云うのではないがもっと大事な事があるというのだ。

先づ

$$[\ ]:\ 式 \longrightarrow \text{boolean value}$$

everywhere operator [ ] を定義する。

式が domain をもつ変数に依存するときは domain のすべての点で true のとき, true, そうでなければ false と定める。

又　$[X] \equiv true$ のとき　$[X]$ と略記することにする。

次に function f について Leibniz's rule に注意する

$$[X \equiv Y] \Rightarrow [f.X \equiv f.Y]$$

これは boole 値を変えない変換が function によって指導されることを示している。つまりこれは function と ≡ の関係を与える公理である.

Leibniz's rule の使い方はいろいろあるが, それに合わせて証明の表記法は工夫される。（第7回の報告を参照のこと）

例えば

$$\begin{array}{ll} & [f.X \equiv f.Y] \\ \Leftarrow & \{\text{Leibnitz's rule}\} \\ & [X \equiv Y] \end{array}$$

$$\begin{array}{ll} & [f.X \equiv f.Y] \\ = & \{\text{hint why } [X \equiv Y];\ \text{Leibniz's}\} \\ & true \end{array}$$

$$\begin{array}{ll} & f.x \\ = & \{\text{hint why } [x = y];\ \text{Leibniz's rule}\} \\ & f.y \end{array}$$

など.

boole値を変えない変換を考えたいのだから
≡中心主義の外には Leibniz's rule があればそれで充分なのである。
≡は equality の性質と operator の性質は兼ね備えているから，時と場合に応じて立場を変えて解釈してかまわない。この様な二重性は極めて多産である．これは自由である。

≡は operatorとして associative 且つ symmetric である

$$[\quad ((X \equiv Y) \equiv Z) \equiv (X \equiv (Y \equiv Z))]$$

$$[\quad (X \equiv Y) \equiv (Y \equiv X)]$$

左右の単位元の存在をたしかめ，それを trueと名づける。
即ち

$$(*) \qquad [X \equiv true \equiv X]$$

を公理として導入する．

$[X \equiv X]$ などは証明される．

例えば証明は次の通り

$$\begin{aligned}
& [X \equiv X] \\
= & \{ [X \equiv (true \equiv X)] \} \\
& [X \equiv true \equiv X] \\
= & \{ (*) \} \\
& true
\end{aligned}$$

q.e.d

少し解説すると　f として　$f.Z = [X \equiv Z]$　を考える

すると　$[X \equiv X]$　は　$f.X$

$[X \equiv true \equiv X]$　は　$f.(true \equiv X)$

にあたるが

$$[X \equiv (true \equiv X)]$$ だから

fX と $f(true \equiv X)$ は等しい．

即ち Leibniz's rule

$$[X \equiv (true \equiv X)] \Rightarrow [f.X \equiv f.(true \equiv X)]$$

が使われているのである．

証明の表記法については オ7日 を参照のこと．

disjunction は symmetric, idempotent, distribute over $\equiv$ で導入され characterize される．

$$[(X \vee Y) \equiv (Y \vee X)], \quad [X \vee X \equiv X]$$

$$[X \vee (Y \equiv Z) \equiv (X \vee Y \equiv X \vee Z)]$$

conjunction は

$$[X \wedge Y \equiv X \equiv Y \equiv X \vee Y]$$

で導入され，characterize される．

これは Golden rule と呼ばれ 美しく生産性の高い公式である．

かくして大学初年級の数学のカリキュラムは例えば
[集合と論理]といった入門講義は
論理のところが、≡ equivalence 中心主義と Leibniz's rule
を要に書きかえられて、
ついでに云っておくが集合は ZFC の foundation axiom
を anti-fundation axiom に変更して、つまり
Aczel's 集合論で書きかえて、循環構造を扱える
ように、hyperset を土台に話を進めるのが自然で
あろう。これは又別の機会に論じよう。