愛知教育大学技術教育研究，２，pp.19～24，March，2016

情報に関する技術の理解を目的とした数理論理学の導入

Lessons of Mathematical Logic for Technology Students to Deep Understanding of IT

鎌田 敏之
愛知教育大学技術教育講座
Toshiyuki Kamada
Department of Technology Education, Aichi University of Education

キーワード：数理論理学，情報に関する技術，計算の原理，プログラミングの基礎
Keywords：Mathematical Logic, Information Technology, Primer of Computing

1. はじめに

本稿の目的は，中学校技術・家庭科 技術分野における教員養成必修科目として，数理論理学に基礎を置く演習形式の授業を設置したことの効果について，授業設計とそれに対する受講生の状況に基づき検証を行うことにある。対象とする授業は，本学技術専攻必修科目である「基礎情報技術」であり，平成 27 年度後期 3 年生を対象に実施した結果を報告する。

数理論理学（Mathematical Logic）は，19 世紀後半以後数学者らにより行われてきた，形式論理（Formal Logic）の枠組みを用いて数学的概念を記述する数学基礎論のひとつである。数理論理学はコンピュータサイエンスの理論（Theoretical Computer Science）を支える重要な学問分野であるとともに，以下のような，中学校技術・家庭科及び高等学校情報科の教科書で扱われている内容に対しても，簡潔かつ明快な説明を与える存在である。したがって，この内容を扱う科目は，教員養成における「情報に関する技術」として重要であると考えている。

(1) 現代のコンピュータは論理回路を基礎として構成されており，二進法が計算に用いられる。
(2) コンピュータは，プログラム内蔵方式（Stored Program）であるノイマン型で構成されている。
(3) プログラムは書かれた通りに逐次実行される。

これらは，それぞれ(1)はブール代数，(2)はチューリングマシン，(3)は計算可能性とチューリング完全 に関わるが，いずれも古典論理（命題論理，一階述語論理）の範囲で十分説明できる。

数理論理学は数学の一分野であり，高等学校で学ぶ数学の知識や技能の延長として位置づけることができる。したがって，大学生にとって，古典論理の入門的な内容は，これまで親しんできた紙と鉛筆を用いた記号操作（等式変形など）の演習によって学ぶことは十分に可能である，というのが筆者の立場である。その上で，実際の情報技術における電子回路，及びソフトウェアとしてどう実現され，応用されているかを具体的に示すことが，中学校技術・家庭科の教員養成の授業設計の上で必須であると考える。

2. 授業設計

平成 27 年度に実施した授業「基礎情報技術」の内容は表 1 の通りである。授業内容は主に教科書 1)を参考に，他の関連する資料類を組み合わせ，再構成した。

表 1 「基礎情報技術」の内容

時	内容
1	位取り記数法と二進法
2	二進法の演算と補数
3	二進法における小数
4	命題
5	命題論理
6	命題論理式の簡単化
7	標準形と真理値表
8	ブール代数の公理系
9	ブール関数と演算表
10	ブール関数の簡単化と標準系
11	カルノー図によるブール関数の簡単化
12	組合せ論理回路

13	複雑な組合せ論理回路
14	順序論理回路
15	順序論理回路の応用−コンピュータ

15回の授業は，4つの大きなモジュールからなる。それは，(1)「数について」(2)「命題と命題論理」(3)「代数学とブール代数」(4)「論理回路による計算」である。

2.1. 数について

このモジュールで意識したことは，数値とその表現が別であることの強調である。*n* 進法表現と基数 *n* の変換については高等学校情報科の教科書でも取り上げられているが，専ら十進法と二進法との相互変換「方法」に焦点が当てられており，より本質的な，「同じ値を異なる表現で記述している」ことについての言及はない。これに加え，負の値を表す補数表現についても「補数がある」ことと，絶対値が同じ正の値を補数表現で負の値にする「手順」のみが示されているため，高校生にとって二進法は直感的でない，なにか特別なものであるような誤解を持った学習がなされている懸念がある。その他，情報に関する技術との関連では，コンピュータで扱うことができる数値は有限の桁数に限られていることが特徴であり，そこから負の値の表現は有限桁であることが利用され，また小数を扱う場合を含め，有限桁数であることが計算誤差の原因のひとつであることの理解も重要である。

そこで，いわゆる「２進数」（二進法で表記された数値に対する慣用表現）は特別な数でもなければ特別な演算が必要なものでもなく，小学校から触れてきた数の，ある特定の表現でしかなく，演算も同じ筆算ができることを示した。またそれに先立ち，これまで学習してきた筆算は「位取り記数法」という表現によって成立していることを，ローマ数字や漢数字での表記を例にし，和算でも独自の位取り記数法によって筆算を実現してきたことを示した。十進法以外の例として，日常には時間を表す際に60進法や12進法が用いられていることにも触れ，基数変換では情報技術でよく利用される8進法と16進法を加え，16進法によって Web などで色を指定する場合があることにも触れた。

さらに，このモジュールで，「すべてはルール（規則）である」ということを強調し，以後の学習で基本となる考え方であることを指摘した。数そのものが，人が概念を共有し考え方を円滑に伝えるための取り決めであり，数値の表現も一定の規則によって成立していること，位取り記数法という共通の規則のなかで基数のみが異なることにより，「*n* 進法」表記が定まることを述べた。

次に，二進法における「補数」の概念を説明するために，集合における補集合を想起させるよう，ベン図を用いた説明を行った。本来，数は空間であり，それは集合として表現されるが，高等学校までの学習では明言されていない。ただし，自然数，整数，実数，複素数のように集合が大きくなり，包含関係にあることが示唆されている。

コンピュータにおける数値の補数表現は，数値が有限の桁で表現されることを利用し，有限集合の元を2分し，最大絶対値からの差によってある値の負値を定めるものとして，任意の基数表現で成り立つ定義を示した。まず，十進法の場合を用いて，3桁で表現可能な1000個の表現（0000〜999）について，999からの距離（差）によって999を負の 0，500を負の499とする偽補数と，負の0をなくし，絶対値の等しいある正の値と対応する負の値の和が0になる真補数があることを先に導入して具体例を演習したのち，二進法の偽補数である1の補数と真補数である2の補数へと進むことで，コンピュータのメモリが扱う{0,1}の2つの状態だけで数値を表現するにあたり，符号ビットを利用した際に生じる演算時の場合分けを避けられる利便性が納得しやすくなるよう工夫した説明を行った。また同時に，減算を負の値である補数との加算となることから，補数表現では四則のうち減算も加算によって実現できることを示した。

さらに，有限桁数であることから，四則演算の結果桁あふれが発生する可能性と，桁あふれの結果，正の値の加算の結果が負の値になったり負の値の減算の結果が正の値になったりすることの危険性，すなわちバグの原因であり，これを悪用した「Integer Overflow」脆弱性についても考えさ

せた。このように，コンピュータ・セキュリティの問題であるプログラムの脆弱性は，主に境界条件から発生するものであり，単純な数値表現のルールが含む制約を見落とすことからも生じ，単純であるからこそ見落とされやすいことを指摘することで，情報モラル教育で抽象的に説明される「プログラムの欠陥」の語と，教科書で概念的に描かれる「壁の穴」といった曖昧な記述を，技術的な理解によって，より正確に認識できるよう配慮した。

小数については固定小数点表現と，指数表現を用いる浮動小数点表現について扱い，両者の得失について考えさせるとともに，有効数字の考え方，それによる丸め誤差，桁落ち，浮動小数点演算で特に顕著な情報落ちによる計算誤差に触れ，十進法の 0.1 が二進法では循環小数になり直感と異なること，現在の電卓では分数として値を内部で保持したり，二進化十進法表現を活用したりするなどして，単純な「10÷3×3」の結果が正しく「10」になるなど，人間の直感に反しない工夫がなされていることについても言及した。誤差については，この授業の範囲を超えるが，数値演算における近似計算で生じる，繰り返しを中断することによる計算打切り誤差についても補足した。

2.2. 命題論理

続いて，高等学校までの学習における証明問題に関連する「推論」を扱う内容として，命題論理をモジュールとして配置した。教科書によっては集合論から代数構造に発展してブール代数へと進む構成をとり，古典論理をブール論理との関係で扱い，計算モデルに発展させることで状態機械であるコンピュータの動作を説明していくものがあるが，数学に馴染みのない受講生への配慮と，高等学校までの学習との連続性，抽象性の高い議論の後に順次論理回路を配置することで，受講生が情報技術を学んだという感覚を得らえるという仮定を総合し，命題論理を先に取り上げることとした。

命題論理は数理論理学における証明論に対応し，対象を記号に置き換え，記号操作によって証明を行うものである。ここで，モジュール(1)の「数について」で数値と表現の分離を意識的に扱ったこととの関連，学習を円滑にする効果を生むと考えた。

命題とはなにか，という定義を考えるなかで，日常的な真偽の存在する事柄を記号に置き換える準備を行い，命題論理式で用いられる加法・乗法・否定の演算子を導入しながら，真偽の定まる文章を命題論理式に変換する演習を行った。命題論理式における演算子である加法・乗法，否定，含意，同値は，集合演算における加法・乗法，さらに空集合・普遍集合（全体集合）・補集合及び，集合の性質である交換率・結合律・分配率・吸収律・べき等律・対合律（二重否定）・排中律・矛盾律・ド・モルガンの法則はそのまま命題論理に対応し，ベン図によって表現することもできるため，集合での表現と命題論理の性質との関係をそれぞれ対応付けて説明した。その後，真理値表を導入することで，真理値表を用いて演算の結果が一致する形で両者が等しいことが示されることも述べた。

含意の論理を説明することの困難さはよく語られることであるが，式 $p \to q$ とは恒真の解釈であり，偽になる条件のみを考慮したとき，p が真で q が偽である場合のみが該当するという解釈が成立する「仮定の規則」であることに注意を払った。ここから背理法も p の仮定から q 及び $\neg q$ が同時に導かれることから，矛盾律により $\neg p$ が解釈されるという説明が可能となる。

このように，命題論理は解釈されるものである。その際，命題論理式を簡単化していく等式変換も解釈の過程とみなされる。簡単化には，公理及びそこから導かれる諸性質を利用することになる。解釈という性質上，正解はひとつでなく，簡単化の過程も一様である必然性はない。よって，受講生には「ルールの適用」，すなわち等式変換に当てはまる性質をルールと見て，置き換えを行う（パズルのような）作業であるという説明を行った。ただし，よりエレガントな変換から，よりシンプルな結果が得られることに気を配るよう，例題の解を板書する際に「どの性質を使うのがよいと思う？」「まだいけるかな？」など意識的な発問を行った。

標準形とは，簡単化の一種であり，式の各項が

リテラルの選言「∧」であり，各項を連言「∨」で結ぶ形を連言標準形，逆に各項をリテラルの連言とし，それを選言で結ぶ形を選言標準形と呼ぶ。命題論理では演繹によって証明を行うことが重要であり，そのための推論に必要な形式であるが，ここでは標準形への変形の演習にとどめ，以後のモジュールで扱うブール関数の標準化とそれを用いた組合せ論理回路の設計に向けての準備と位置づけた。

2.3. 代数学とブール代数

さきのモジュールで，命題論理をブール論理により形式化された結果に基いて説明しているため，ブール代数の内容を結果だけ扱ってしまうと，命題論理と記号が違うだけでほぼ同様の公理や性質を説明するのみとなってしまい，混乱が生じることを懸念した。

そこで，代数系（algebra）について，群論に基づく定義，すなわち集合 *A* と *A* に閉じた演算◦との組（*A*;◦）からなる代数と，代数的構造を規定する公理系を用いて，群・環・体・束といった代数系が定義されることを説明し，そこに順序関係を加え，最大元と最小元を持つ代数系（*B*; +, ）がブール束，すなわちブール代数であることをコンパクトに説明した。

これにより，高等学校までに実数 ***R*** での四則演算の性質として学んでいた交換律，分配律，結合律なども，代数系の考え方では「実数体」(***R***; +, ）の定義の一部であることが理解され，ブール代数（*B*; +, ）の名も同様の考え方からきていることが理解されることを期待した。

次に，ブール代数では式は関数として定義でき，変数の組の値の組み合わせにより，命題論理における真理値表と同様の演算表が作成できることを示した。また，ここで初めて同値の否定である「排他的論理和」を演算に加えた。排他的論理和は「変数 *x* と *y* が，それぞれ異なる値をもつかどうか」を調べる役割があり，次のモジュールで扱う組合せ論理回路では重要な役割を果たすものである。

特定の目的を持つ組合せ論理回路を導くために，対応するブール関数を作成できることが，情報技術の理解では重要となる。その手段として，まずブール関数の簡単化を公理や性質というルールの組合せから行う演習を行い，加法標準形と乗法標準系に変換する手順についても演習を行った。そして，任意の演算表から加法標準形に基づきブール関数を導く方法について演習した。

最後に，図表を用いてブール関数を簡単化する方法としてカルノー図を用いる方法と，ブール代数の演算が，定義で用いられた和（OR）・積（AND）･否定（NOT）の 3 つ組だけでなく, NAND（NOT AND すなわち積を否定する演算）単独でも他の演算すべてを表現できること，すなわち「完全系」をなすことを示し，完全系をなす演算には様々なものがあることを紹介し，その導出手順についても説明した。これは例えば NAND 論理回路ひとつがあればあらゆる組合せ論理回路が作成であり，NAND は直列のスイッチとなるトランジスタ（FET）を負論理で配置したものであるから，大規模な論理回路を安価に作成するという実用的な技術に応用される。

2.4. 論理回路による計算

最後のモジュールは，これまで学んだ二値論理関数を電気回路として実現し，コンピュータがどのように実現されているかの一端を知る内容である。

基本の論理回路は論理和（OR）はスイッチを回路のなかで並列に並べたものであり，論理積（AND）は直列に並べたものである。否定（NOT）はトランジスタ（FET）を用いることで論理を反転させた出力を得ることができる。さきのモジュールで学んだ加法標準形を用いたブール関数の導出の手順から，任意の入力に対し任意の出力を得る論理回路を，基本の論理回路を結合した，組合せ論理回路として実現することができる。

具体的な例として，加算回路を取り上げた。1 桁の二進法の値を 2 つ加算する回路は，その結果として，和となる桁はちょうど XOR 論理に相当し，1+1 となる場合に繰り上がりが発生し，繰り上がり出力はこのときのみ 1 を出力することになるため，AND 論理と合致する。複数桁の二進法の値の場合，最下位桁以外では，下の位からの繰り上がりを考慮した，3 つの値の和を行わなければならないが, 和の出力は 3 つの入力のパリティ（奇

偶）の出力であり，繰り上がりの出力は 3 入力の XOR 論理に合致する。このように，論理演算によって数値の演算を実現することができるが，こうした発明は，1930 年代に日本電気の技術者であった中嶋章が「スイッチング理論」として日本語で発表し，クロード・シャノンらが引用する形で展開し，世界に知られた経緯がある。[2)]

加算回路が実現できれば，減算は引く数を 2 の補数表現に変換して加算するだけである。2 の補数とは，入力のすべての桁の論理を反転した後，最下位桁に 1 を加える操作で実現できる。乗算及び除算は原則として，筆算の手順がそのまま利用できるため，加算と減算さえできれば乗算も除算も実現できる。

「計算する機械」としての数値及び論理演算はこうして実現できるが，状態を持つ機械としてのコンピュータにはならない。このような回路を順序論理回路と呼び，最も単純なものとしてフリップフロップ回路が挙げられる。フリップフロップは入力された 1 または 0 の値を保持して出力する回路であり，論理回路による 1 ビットの記憶装置でもある。授業では，基本となる RS フリップフロップ及びクロック信号により入力端子の値を保持する D フリップフロップを解説し，ここからクロックごとに数を数えるカウンタ回路が実現され，メモリに保存されたプログラムのデータを先頭から順に CPU に読み出す順序処理に関連していることを示唆して終了した。

3. 授業の評価

授業は教科書の例題及び演習問題の一部を解きながら，必要な知識・技能を板書により補う形式で行った。

授業における解説で，特に受講生に印象的との反応が得られたのが，命題論理における排中律と矛盾律について，プログラムで条件分岐を行う際に条件設定を「漏れなく」「重複を避けて」行わなければ，意図した分岐動作を実現することができず，ここに誤りが含まれた場合，プログラムも誤った動作となることの指摘であった。授業は 3 年生後期に実施しており，前期にプログラミングの実習授業を済ませているため，課題作成時に，意図をどのようにプログラムに反映させるかを迷った者，予想外の動作に戸惑い相談を受けて修正した経験のある者からの反応が強かった。

他に，数に関するモジュールでの誤差の話題に関心を示す様子が伺われた。有効桁数について，これまで意識を払っていなかったことにあらためて気づいた，との発言もあった。

ブール関数と組合せ論理回路については，数学的要素が少なく手順的に作業が進むため，試験結果をみる範囲では，学習内容を再現することは比較的容易であったようである。ただし，加算器までは再現できるものの，「3 桁の 2 つの値を加算する回路を作成する」問題について，自力で回路を完成させられる者は限られていた。フリップフロップ回路については，電気の授業で学習しているものの，論理から演繹的に導く議論については戸惑いを見せていたため，順序論理回路については試験範囲としなかった。よって，どの程度の理解が及び，情報技術との関連を理解できたかは不明である。

表 2 に，最終試験と授業の各モジュールに対する正答率を示す。それぞれ，(1)二進法の基本的な計算 (2)論理式の簡単化 (3)ブール関数の簡単化と完全系 (4)ブール関数の作成と組合せ論理回路の作図 (4)’ 3 桁の加算器 に対応する。受験者の人数は 13 名であった。

表 2 最終試験の各モジュールと正答率

問題	正答率（%）
(1)	92.3
(2)	84.6
(3)	80.8
(4)	96.2
(4)’	46.2

試験結果からみて，授業で扱った知識と技能に関しては，全体としておおむねよく定着したと考えられる。一方，(4)’は応用問題であり，授業中に半加算器と全加算器の作成までは示したが，それを複数桁の加算器に構成するところまでは概念的に述べたのみで，図を示すことはしなかったことから，半数未満のみが十分な回答をするにとどまり，問題の意図を理解していない作図が目立った。(3)が(2)と較べ正答率が低いが，完全系を示

すためにはすべての演算 8 つについて，特定の演算のみを用いて導出しなければならないが，全部を回答していなかったり，一部を誤ったりした者がいたことが原因であり，回答すべき分量の違いと捉えている。

4. まとめと今後の課題

今回構想し実施した 4 つのモジュールにおいて，中学校及び高等学校学習指導要領が求める範囲に対し，上位に位置する抽象的な概念からの理解を求める内容と，より具体的な電気回路の構成に踏み込んだ内容を加えることで，教員養成課程の大学生として期待する「情報に関する技術を支える考え方と技術」の内容を盛り込む授業を構想し，実施した結果について報告した。

カリキュラムの配列として，親学問である数理論理学及び論理回路の一般的な学習順序に対し，高等学校までの学習内容により近い部分を導入部分に移動させる工夫により，一定の知識・技能の定着が可能になったと考えられる。ただし，数学の演習のように，紙の上で手を動かしながら考える時間が多かったため，得られた知識・技能を応用しようとする段階には至らなかったことが，反省点である。

それを解決するため，演習として問題を解く時間が間延びすることを避け，より情報技術とのつながりを強化する方向に時間配分を調整する手段が考えられる。具体的には，順序論理回路の先にある，CPU でのプログラム実行に関するイメージの形成を目的とした対話的な授業である。ただし，CPU 内での制御機構については学習者のレベルに合わせた多様な水準での取扱いに対する見通しがあるが，記憶装置などメモリ空間については，順序論理回路であるデコーダ回路の知識が必須であり，フリップフロップ回路から演繹的に示す必要があるため，時間内での取扱は難しいと予想される。

抽象的な議論の範囲では，数理論理学の枠組みのうち述語論理について全く扱っていないため，モーダスポネンス (modus ponens) を始めとする推論と，そのカット規則から計算順序の制御といった計算の本質に関わる議論展開が欠けている。しかし，順序論理回路から「コンピュータは書かれたプログラムにしたがって順序動作する」ことは説明可能であり，教員養成の必修授業として適切な内容かについてはさらなる考察が必要と考える。

6. 参考文献

1) 小椋久和・高濱徹行：情報の論理数学入門―ブール代数から述語論理まで―，近代科学社 (1991)

2) 山田昭彦：スイッチング理論の原点を尋ねて―シャノンに先駆けた中嶋章の研究を中心に―，電子情報通信学会 Fundamentals Review, Vol.3, No.4, pp.9-17 (2009)

(2016 年 3 月 26 日受理)