2022年度 数理論理学

講義資料(7)

青戸 等人 (知能情報システムプログラム)

目次

メタ変数

これまで，命題論理式が2通りの方法で表現されてきたことに気がついただろうか？

- 1つは，$\mathtt{P} \land \mathtt{Q} \to \mathtt{P}$のような，命題変数から構成されている，具体的な表現

- 1つは，$A \to B$のような，命題論理式の形だけを規定するための表現

A, Bは実体は命題論理式だが，その具体的な中身は不定，例えば，$\mathtt{P} \to \mathtt{Q}$なのか$\mathtt{P} \land \mathtt{Q}$なのか，固定されていない．従って，$A, B$が指している命題論理式によって，$A = B$($A, B$が同じ命題論理式を指す場合)にも，$A \neq B$($A, B$が異なる命題論理式を指す場合)にもなりうる．

つまり，A, Bは字面が異なるからといって，$0 \neq 1, \mathrm{T} \neq \mathrm{F}, \mathtt{P} \neq \mathtt{Q}$のように，$A \neq B$とすることは出来ない．

一方，A, Bはアルファベットの1文字という意味では，$\mathtt{P}$や$\mathtt{Q}$と何ら変わるものではない．ある種の“変数”ともいえる．

実は，“A”や“B”は議論や説明を行うというメタな立場で用いている変数である．言い替えると，議論をしている人が使っている変数である．これらは，議論の対象となる(命題論理の)世界にある命題変数とは異なる．

“A”や“B”のような，メタな立場で用いている変数をメタ変数とよぶ．

メタ変数 (2)

同様なことが命題変数についてもいえる．つまり，次の2種類は区別するべき：

- 命題変数
- 命題変数のメタ変数

この講義では，$P, Q, \ldots$を，命題変数のメタ変数として用いる．

つまり，

- $\mathtt{P}_0, \mathtt{P}_1, \ldots$や$\mathtt{P}, \mathtt{Q}, \ldots$は命題変数
- $P, Q, \ldots$は命題変数のメタ変数

メタ変数の役割を見るために，命題変数集合$\mathtt{Var}$および命題論理式集合$\mathtt{Prop}$の定義($\top$と$\bot$が追加されている$\cdots$講義資料(3)p.20参照) を示しておく．

定義 7.1. [再掲] 命題変数集合$\mathtt{Var}$および命題論理式集合$\mathtt{Prop}$を以下のように与える：

$$
\begin{array}{rcl}
P, Q \in \mathtt{Var} & ::= & \mathtt{P} \mid \mathtt{Q} \mid \cdots \\
A, B \in \mathtt{Prop} & ::= & P \mid \top \mid \bot \mid (\neg A) \\
 & & (A \land B) \mid (A \lor B) \mid (A \to B) \mid (A \leftrightarrow B)
\end{array}
$$

例．$P \to Q$は$\mathtt{P} \to \mathtt{P}$かもしれないし，$\mathtt{Q} \to \mathtt{P}$かもしれないが，$\mathtt{Q} \land \mathtt{P} \to \mathtt{P}$ではない．$A \to B$は$\mathtt{P} \to \mathtt{P}$かもしれないし，$\mathtt{Q} \land \mathtt{P} \to (\mathtt{Q} \to \mathtt{P})$かもしれない．

メタ数学 (1)

命題論理の“$\land, \lor, \to, \leftrightarrow, \neg$”と，説明や証明で用いる「かつ，または(場合分け)，ならば($\Rightarrow$)，同値($\Leftrightarrow$)，〜でない(否定)」も，命題変数とメタ変数と同じ意味で，使うレベルが異なる．

しかし，その意味は同じ．つまり，証明で用いる(数学に用いる)「かつ，または(場合分け)，ならば($\Rightarrow$)，同値($\Leftrightarrow$)，〜でない(否定)」による解釈は，命題論理で用いる“$\land, \lor, \to, \leftrightarrow, \neg$”の解釈と同じである．

なお，この講義では，$\Longrightarrow$と$\Rightarrow$の意味は同じであり，使い方で区別しているだけである．($\Longleftrightarrow$と$\Leftrightarrow$も同様．)

実際，以下が成立する：

$$
\begin{array}{l}
[\![A \land B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{かつ}[\![B]\!]_v = \mathrm{T}) \\
[\![A \lor B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{または}[\![B]\!]_v = \mathrm{T}) \\
[\![A \to B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{ならば，}[\![B]\!]_v = \mathrm{T}) \\
[\![A \leftrightarrow B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{であるとき，} \\
\qquad\qquad\qquad\qquad \text{そのときに限り，}[\![B]\!]_v = \mathrm{T}) \\
[\![\neg A]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{，ではない})
\end{array}
$$

$$
\begin{array}{l}
[\![A \to B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T} \Longrightarrow [\![B]\!]_v = \mathrm{T}) \\
[\![A \land B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T}\text{，}[\![B]\!]_v = \mathrm{T}) \\
[\![A \leftrightarrow B]\!]_v = \mathrm{T} \Longleftrightarrow ([\![A]\!]_v = \mathrm{T} \Longleftrightarrow [\![B]\!]_v = \mathrm{T})
\end{array}
$$

このような意味で，数理論理学をメタ数学(超数学)とよぶことがある．なお，メタ言語は自然言語(日本語)であり，自然言語では括弧は通常使わないことに注意．

メタ数学 (2)

対象となる論理で用いられる言葉と，数学で用いられる言葉の対応を示す．

論理	数学
(省略方法の約束)	定義
論理式	命題 (真偽は問わないで使う場合)
トートロジー/定理	定理/補題/命題 (使い方で区別する)
$\wedge, \vee, \to, \leftrightarrow, \neg$	かつ，または，ならば ($\Rightarrow$)，同値 ($\Leftrightarrow$)，〜でない
$\cong$による同値変形	$\Longleftrightarrow$による同値変形
証明図	証明
$\forall, \exists$	任意の〜について，〜が存在して

※ $\forall, \exists$については，述語論理で学習．

証明図から (数学の) 証明へ

数学の証明は，この後で学習する述語論理に基づいている．このため，命題論理の推論だけでは，数学の証明を記述することはできない．しかし，述語論理は命題論理の拡張であり，命題論理の推論は述語論理でもそのまま用いることが出来る．したがって，どのように，自然演繹法の推論が数学の証明に用いられているかを知っておくことは，自然演繹法の推論を実際の応用する上では非常に重要．

そこで，以下では，命題論理の証明図を，数学の証明で使われる形へ近づけたものとして，“擬似証明コード” というものを考える．

目次

擬似証明コード

証明図は2次元的な図の形をしているが，数学の証明は自然言語で記述され，先頭から順番に書かれる1次元的な “文章” になっている．

計算機のプログラムを思い出してみよう．プログラムでは，先頭から順番に，構造的な手続きが記述されている．例えば，if文では，条件節+then節+else節によって構成されているが，これらは図ではなく順番に記述されている．プログラムの実行では，構文解析というコンパイラの処理過程の一部で，プログラムの構造が解析される．

そこで，まず，証明図を1次元的な記述に変換してみよう．我々は，これを擬似証明コードとよぶ．

assume キーワード

証明図を構成する最初のステップは，ある命題論理式Aを用いて，

$$[A]$$

という図を作ることであった．

これを，擬似証明コードでは，

```
assume A
```

と記述する．ここで，assumeは，擬似証明コードで用いる特別な予約語である．

なお，四角の囲みは，擬似証明コードであることの表わす説明のためのもので，擬似証明コードの一部ではない．

ラベルによる命題論理式の参照

証明図では，2次元的に，推論の前提と結論を水平線の上下で示すことが出来た．

$$\frac{B \quad C}{A}$$

しかし，1次元的な擬似証明コードでは，そのようなことが出来ない．そこで，ラベルというものを導入する．

擬似証明コードで用いる命題論理式には，すべて$\ell : A$という形で，ラベルℓを付加してよい．

実際には，ラベルには，どのような文字列を用いてもよいが，以下では，自然数を用いることにする．

擬似証明コードにおける推論

擬似証明コードの基本的な文は，assume文に加えて，推論ステップを表わす以下の形の文である：

```
from ℓ1 ℓ2 ··· ℓn have ℓ : A by rule
```

ここで，

- $\ell_1\ \ell_2\ \cdots\ \ell_n$は仮定の命題論理式ラベルの列
  (ラベルの列の順番は任意でよい．)
- from, have, byは擬似証明コードの予約語
  ($n = 0$のときfromは省略．)
- Aは結論の命題論理式
- (必要がなければ) ℓ :は無くてもよい
- *rule*は，用いた自然演繹法の推論規則名

擬似証明コードの例 (1)

例えば，証明図

$$\frac{[\mathtt{P} \to \mathtt{Q}] \quad [\mathtt{P}]}{\mathtt{Q}} \to\mathrm{E}$$

は，擬似証明コードでは，以下のように記述する．

```
assume 1 : P → Q
assume 2 : P
from 1 2 have Q by →E
```

擬似証明コードにおける仮定の除去

証明図では，推論規則と仮定の肩に自然数を付加することで，仮定の除去を表わした．

擬似証明コードでは，仮定に付いているラベルを推論規則において指定することで，仮定の除去を表わす．

from $\cdots$ have $\ell : A$ by $\boldsymbol{rule}[\ell_1, \ldots, \ell_n]$

$\boldsymbol{rule}$が仮定を除去できる自然演繹法の推論規則であり，除去する仮定のラベルが$\ell_1, \ldots, \ell_n$であるとき，$\boldsymbol{rule}$の直後に$[\ell_1, \ldots, \ell_n]$を記述して，これらの仮定が除去されたことを示す．

擬似証明コードの例 (2)

例えば，証明図

$$\dfrac{\dfrac{[\mathtt{P} \to \mathtt{R}] \qquad \dfrac{[\mathtt{P} \land \mathtt{Q}]^1}{\mathtt{P}}\land\mathrm{E}}{\mathtt{R}}\to\mathrm{E}}{\mathtt{P} \land \mathtt{Q} \to \mathtt{R}}\to\mathrm{I}^1$$

は，擬似証明コードでは，以下のように記述する．

```
assume 1 : P ∧ Q
from 1 have 2 : P by ∧E
assume 3 : P → R
from 2 3 have 4 : R by →E
from 4 have P ∧ Q → R by →I[1]
```


using キーワード

次に，実際の，自然言語での証明に近づけるために，擬似証明コードに，いくつかの柔軟な機構を導入していく．

- from $l_1 \;\cdots\; l_n$ have $\ell : A$ の代わりに
  have $\ell : A$ using $l_1 \;\cdots\; l_n$ を用いてよい．
- 仮定を適当に分割し，fromとusingの両方を用いてよい．

```
assume 1 : P ∧ Q
have 2 : P using 1 by ∧E
assume 3 : P → R
from 3 have 4 : R using 2 by →E
from 4 have P ∧ Q → R by →I[1]
```


this キーワード

fromの後では，ラベルの代わりに，予約語thisを用いることが出来る．thisは，直前の行の命題論理式を指す．

```
assume 1 : P ∧ Q
from this have 2 : P by ∧E
assume 3 : P → R
from this have R using 2 by →E
from this have P ∧ Q → R by →I[1]
```

thisを使うことで，直後にしか参照されない命題論理式のラベルを省略することが出来る．

then キーワード

from thisの代わりにthenを用いる．

```
assume 1 : P ∧ Q
then have 2 : P by ∧E
assume 3 : P → R
then have R using 2 by →E
then have P ∧ Q → R by →I[1]
```


演習 7.2. 以下の証明図に対応する擬似証明コードを書け．

$$\dfrac{\dfrac{\dfrac{[\neg\mathtt{Q}]^2 \qquad \dfrac{[\mathtt{P} \to \mathtt{Q}] \qquad [\mathtt{P}]^1}{\mathtt{Q}}\to\mathrm{E}}{\bot}\neg\mathrm{E}}{\neg\mathtt{P}}\neg\mathrm{I}^1}{\neg\mathtt{Q} \to \neg\mathtt{P}}\to\mathrm{I}^2$$

演習 7.2. 以下の証明図に対応する擬似証明コードを書け．

$$\dfrac{\dfrac{\dfrac{[\neg\mathtt{Q}]^2 \qquad \dfrac{[\mathtt{P} \to \mathtt{Q}] \qquad [\mathtt{P}]^1}{\mathtt{Q}}\to\mathrm{E}}{\bot}\neg\mathrm{E}}{\neg\mathtt{P}}\neg\mathrm{I}^1}{\neg\mathtt{Q} \to \neg\mathtt{P}}\to\mathrm{I}^2$$

```
assume 1 : P → Q
assume 2 : P
then have 3 : Q by →E
assume 4 : ¬Q
then have ⊥ using 3 by ¬E
then have ¬P by ¬I[2]
then have ¬Q → ¬P by →I[4]
```


hence キーワード

then have A の代わりにhence Aを用いる．

```
assume 1 : P ∧ Q
hence 2 : P by ∧E
assume 3 : P → R
hence R using 2 by →E
hence P ∧ Q → R by →I[1]
```

and キーワード

命題が(同じ仮定を使って)同じ推論から導かれるときは，and で複数(順序不問)の命題を導くことができる．

```
assume 1 : P ∧ Q
from 1 have 2 : P by ∧E
from 1 have 3 : Q by ∧E
from 2 3 have Q ∧ P by ∧I
```

2,3行目を1つにする：

```
assume 1 : P ∧ Q
hence 2 : P and 3 : Q by ∧E
from 2 3 have Q ∧ P by ∧I
```


さらに，省略すると…

```
assume 1 : P ∧ Q
hence 2 : P and 3 : Q by ∧E
from this have Q ∧ P by ∧I
```

```
assume 1 : P ∧ Q
hence P and Q by ∧E
hence Q ∧ P by ∧I
```

演習 7.3.　以下の証明図に対応する擬似証明コードを書け．

$$\dfrac{\dfrac{\dfrac{\dfrac{[P\to Q\to R]^2 \quad \dfrac{[P\land Q]^1}{P}\land E}{Q\to R}\to E \qquad \dfrac{[P\land Q]^1}{Q}\land E}{R}\to E}{P\land Q\to R}\to I^1}{(P\to Q\to R)\to P\land Q\to R}\to I^2$$

$$\dfrac{\dfrac{\dfrac{\dfrac{[P\to Q\to R]^2 \quad \dfrac{[P\land Q]^1}{P}\land E}{Q\to R}\to E \qquad \dfrac{[P\land Q]^1}{Q}\land E}{R}\to E}{P\land Q\to R}\to I^1}{(P\to Q\to R)\to P\land Q\to R}\to I^2$$

解答例：

```
assume 1 : P → Q → R
assume 2 : P ∧ Q
hence 3 : P and 4 : Q by ∧E
from 1 3 have Q → R by →E
hence R using 4 by →E
hence P ∧ Q → R by →I[2]
hence (P → Q → R) → P ∧ Q → R by →I[1]
```


連続する同じ文の簡略化

- 連続する assume 文は，and で繋いで，一緒にしてよい．
- 連続する →E は，一緒にしてよい．
- 連続する →I は，一緒にしてよい．

先の解答例の簡略化：

```
assume 1 : P → Q → R and 2 : P ∧ Q
from 2 have 3 : P and 4 : Q by ∧E
hence R using 1 by →E
hence (P → Q → R) → P ∧ Q → R by →I[2, 1]
```


trivially キーワード

仮定から導く命題が全く同じ形のときは，trivially で導く．

$$\dfrac{\dfrac{[P]^2}{Q\to P}\to I^1}{P\to Q\to P}\to I^2$$

```
assume 1 : P and 2 : Q
hence P trivially
hence P → Q → P by →I[2, 1]
```

```
assume 1 : P
hence P trivially
hence P → Q → P by →I[1]
```


目次

後づけ証明

示す式と証明の順序を逆にすることが出来る．show で示す命題を提示し，その下にその命題の証明を書く．

```
assume 1 : P → R
show P ∧ Q → R as follows:
  assume 2 : P ∧ Q
  hence P by ∧E
  hence R using 1 by →E
  hence claim by →I[2]
hence (P → R) → P ∧ Q → R by →I[1]
```

インデントされている部分が，命題の証明．示すべき命題は，claim と参照されている．


```
assume 1 : P → R
show P ∧ Q → R as follows:
  assume 2 : P ∧ Q
  hence P by ∧E
  hence R using 1 by →E
  hence claim by →I[2]
hence (P → R) → P ∧ Q → R by →I[1]
```

(1) 示す命題を show と as follows: で囲む．

(2) その直後に，その命題の証明をインデントをつけて書く．後付け証明は，示す命題を claim と導いて終わる．

(3) 後付け証明の中では，assume する仮定と除去する仮定を一致させる．

∨E規則(場合分け)による証明 (例)

$$\dfrac{[\mathtt{P} \lor \mathtt{Q}] \qquad \dfrac{[\mathtt{P} \to \mathtt{R}] \quad [\mathtt{P}]^1}{\mathtt{R}}\,{\to}\mathrm{E} \qquad \dfrac{[\mathtt{Q} \to \mathtt{R}] \quad [\mathtt{Q}]^1}{\mathtt{R}}\,{\to}\mathrm{E}}{\mathtt{R}}\,{\lor}\mathrm{E}^1$$

```
assume 1 : P ∨ Q and 2 : P → R and 3 : Q → R
show R as follows:
  assume 4 : P
  hence R using 2 by →E
  next
  assume 5 : Q
  hence R using 3 by →E
  done
  hence claim using 1 by ∨E[4, 5]
```


↔I規則による証明 (例)

$$\dfrac{\dfrac{\dfrac{[\mathtt{P} \land \mathtt{Q}]^1}{\mathtt{Q}}\,{\land}\mathrm{E} \quad \dfrac{[\mathtt{P} \land \mathtt{Q}]^1}{\mathtt{P}}\,{\land}\mathrm{E}}{\mathtt{Q} \land \mathtt{P}}\,{\land}\mathrm{I} \qquad \dfrac{\dfrac{[\mathtt{Q} \land \mathtt{P}]^1}{\mathtt{P}}\,{\land}\mathrm{E} \quad \dfrac{[\mathtt{Q} \land \mathtt{P}]^1}{\mathtt{Q}}\,{\land}\mathrm{E}}{\mathtt{P} \land \mathtt{Q}}\,{\land}\mathrm{I}}{\mathtt{P} \land \mathtt{Q} \leftrightarrow \mathtt{Q} \land \mathtt{P}}\,{\leftrightarrow}\mathrm{I}^1$$

```
show P ∧ Q ↔ Q ∧ P as follows:
  assume 1 : P ∧ Q
  hence P and Q by ∧E
  hence Q ∧ P by ∧I
  next
  assume 2 : Q ∧ P
  hence Q and P by ∧E
  hence P ∧ Q by ∧I
  done
  hence claim by ↔I[1, 2]
```


∨E規則と↔I規則による証明

∨E規則と↔I規則による推論では，上にあるそれぞれの証明図で除去する仮定が異なる．このため，以下のように，後付け証明を用いる．

(1) as follows:の後に，まず，真ん中の証明図(∨E規則の場合，↔I規則の場合は左側)に対応する擬似証明コードを記し，nextキーワードで終了．

(2) 右側の証明図に対応する擬似証明コードを記し，doneキーワードで終了．

(3) ∨E規則や↔I規則により，claimを導く．

注意:それぞれの証明図に対応するブロック内では，assumeの使い方に次ページの約束がある．

約束：それぞれの証明図に対応するブロック内でassumeできるのは，∨E規則や↔I規則で除去される仮定のみ．

… ブロックの前にassumeしておく必要がある．

以下は間違い：

```
assume 1 : P ∨ Q
show R as follows:
  assume 2 : P → R
  assume 4 : P
  hence R using 2 by →E
   ... (!!仮定 2 が除去されていない!!)...
  next
  assume 3 : Q → R
  assume 5 : Q
  hence R using 3 by →E
   ... (!!仮定 3 が除去されていない!!)...
  done
  hence claim by ∨E[4, 5]
```


∨E規則(場合分け)による証明 (改良版)

∨Eによる証明は，以下のように，よりわかりやすく書くことが出来る．

```
assume 1 : P ∨ Q and 2 : P → R and 3 : Q → R
show R as follows:
  distinguish cases using 1
  case P
    hence R using 2 by →E
  case Q
    hence R using 3 by →E
  hence claim by ∨E
hence (P → R) → (Q → R) → P∨Q → R by →I[1, 3, 2]
```


↔I規則による証明 (改良版)

↔Iによる証明は，以下のように，よりわかりやすく書くことが出来る．

```
show P ∧ Q ↔ Q ∧ P as follows:
  (=>)
    assume 1 : P ∧ Q
    hence P and Q by ∧E
    hence Q ∧ P by ∧I
  (<=)
    assume 2 : Q ∧ P
    hence Q and P by ∧E
    hence P ∧ Q by ∧I
  hence claim by ↔I[1, 2]
```


目次

擬似証明コードから数学の証明へ

擬似証明コードはきちんとした形式が決まっているが，数学の証明は自然言語で記述され，記述の自由度が高い．逆に自然言語で記述することから，曖昧さを排除して，明確に意図を伝える工夫が必要になる．また，数学の証明では，自然演繹法以外の推論でも正しい推論であれば使ってよい．

前述したように，数学の証明は述語論理に基づいているので，ここで説明した擬似証明コードは，数学の証明とはまだ少しギャップがある．しかし，今後，出てくるさまざまな証明では，擬似証明コードと対応している部分が各所に観察できるはずである．ぜひ，それを見てとって欲しい．

我々の擬似証明コードは，もう少し手を加えれば，実際の数学の証明に用いられている命題推論の形になる．

実際の数学の証明に用いられている命題推論との違いは以下の通り：

(1) どのような推論規則を使ったかは明示しない．
(2) 各予約語は，対応する日本語にする．
(3) 論理記号は，対応する日本語にする．
(4) 最低限のラベルしか使わない．
(5) 正しい文章にする．

つまるところ，数学の証明は，論理的な推論を身につけた人が，自然に読めるように書くのが，マナーということ．

擬似証明コードの予約語に対応する日本語

以下では，擬似証明コードの主な予約語に対応する日本語を示しておく．

擬似証明コードで用いた予約語のほとんどは，実は，その場面場面で用いられる代表的な英単語である．英語でも日本語でも，文章が単調になるのを避け，読みやすく(頭に入りやすく)するために，さまざまな言い方を用いる．

`assume` A
　Aであると仮定する，Aとする

`have` A
　Aが成立する，Aが成り立つ，Aが得られる

`hence`
　従って，ゆえに，よって，すると，すなわち，以上より

`from`/`using`
　〜より，〜から，〜を使って，〜を用いて

`show` A
　Aを示す，Aを証明する

`as follows`
　以下の通りに

`next`
　次に

`have claim`, `done`
　以上で示された

`trivially`
　自明に

`distinguish cases`
　場合分けを行う

`case` A
　Aの場合，Aのとき

まとめ

- 数理論理学 ＝ メタ数学
  - メタ変数
  - 命題結合子と命題結合の対応
  - 論理の言葉と数学の言葉の対応
- 擬似証明コード
  - 証明図から，数学の証明へ
  - 数学の証明における命題推論の使われ方
  - `assume`, `from`, `have`, `by`, `using`
  - `this`, `then`, `hence`, `trivially`
  - `show`, `claim`, `next`, `done`
  - `distinguish cases`, `case`, `(=>)`, `(<=)`