数理論理学入門

神戸大学 田村直之 (tamura@kobe-u.ac.jp)

1　数理論理学とは

代数を用いると，数の計算を記号的 (形式的) に行うことができる．例えば，以下の鶴亀算の問題を考えてみる．

「鶴と亀があわせて 5 匹いる．足の本数の合計は 16 本である．鶴と亀はそれぞれ何匹いるか.」

(1) 問題の記号化 (形式化):
鶴を x 匹，亀を y 匹とすると，問題は連立方程式 ($x + y = 5$, $2x + 4y = 16$) で表される．

(2) 形式化した問題の解を求める:
この連立方程式を解くと，解 ($x = 2$, $y = 3$) を得る．

(3) 解を解釈し，最初の問題の答えを得る:
答えは「鶴は 2 匹で亀は 3 匹」であることがわかる．

数理論理学(Mathematical Logic) を用いると，論理的な推論を記号的 (形式的) に行うことができる．数理論理学は，数学基礎論における研究分野の一つであり，**記号論理学**(Symbolic Logic) とも呼ばれる

例えば，以下の三段論法の推論を考えてみる．

「人は必ず死ぬ．ソクラテスは人である．したがって，ソクラテスは必ず死ぬ.」

(1) 問題の記号化 (形式化):
「人である」を述語 H,「必ず死ぬ」を述語 M，ソクラテスを定数 s で表すと，問題中の最初の二つの文はそれぞれ $\forall x(H(x) \Rightarrow M(x))$ および $H(s)$ という論理式に記号化できる．

(2) 形式化した問題の解を求める:
これらの論理式から $M(s)$ という論理式を記号的 (形式的) に導き出すことができる．

(3) 解を解釈し，最初の問題の答えを得る:
$M(s)$ を解釈して「ソクラテスは必ず死ぬ」ことがわかる．

ステップ (2) では，記号の元々の意味を考える必要はないことに注意してほしい．連立方程式の場合は，x が鶴で y が亀であることを知らなくても (意識しなくても) 解を求めることができ，同様に記号論理の場合にも H, M, s が何を表しているか知る必要はない．

数理論理学 (記号論理学) は，このような正しい (何を表しているかによらず常に正しい) 推論を行う方法を提供してくれる．

$$
\begin{array}{ccccc}
 & & \text{形式的推論} & & \\
 & \textbf{論理式} & \Longrightarrow & \textbf{解 (定理)} & \\
\text{記号化 (形式化)} & \Uparrow & & \Downarrow & \text{解釈} \\
 & \textbf{問題} & \longrightarrow & \textbf{答} &
\end{array}
$$

1.1 数理論理学の計算機科学への応用

- プログラミング言語
  - 型理論 (関数型言語，XML)
  - 論理型言語 (Prolog)
- システム仕様記述，検証
- 人工知能 (Artificial Intelligence)

1.2 数理論理学の体系

数理論理学は，どのような推論を必要とするのかに応じて，様々な**体系**がある．

- 古典論理 (Classical Logic)
  - 命題論理 (Propositional Logic)
  - 一階述語論理 (First Order Predicate Logic)
  - 二階述語論理
- 非古典論理
  - 直観主義論理
  - 様相論理 (時相論理)
  - 線形論理
  - 多値論理 (三値論理，ファジィ論理)

2 命題論理 (propositional logic)

2.1 構文 (syntax)

通常の数式が，定数，変数および演算子から構成されるように，命題論理の論理式も命題定数，命題変数および論理結合子から構成される．

本節では，どのような式を命題論理式として使用するのか，すなわち命題論理式の**構文**(syntax) を定める．

2.1.1 命題定数 (propositional constant)

命題論理における**命題定数**は以下の二種類である．

- $\bot$
- $\top$

命題定数 $\perp$ は**偽** (false) を表し，$\top$ は**真** (true) を表す．

2.1.2 命題変数 (propositional variable)

命題変数(propositional variable) は，真か偽のどちらかの値を取る変数であり，ここでは以下のような記号を用いる．

$$p,\ q,\ r,\ p_1,\ q_1,\ r_1,\ \ldots$$

命題変数は，**ブール変数**(Boolean variable) と呼ばれることもある．

2.1.3 論理結合子 (logical connective)

ここでは，以下の**論理結合子**(logical connective) を用いる．

- $\neg$ (not, 否定, negation)
- $\wedge$ (and, 論理積, 連言, conjunction)
- $\vee$ (or, 論理和, 選言, disjunction)
- $\Rightarrow$ (implies, 含意, implication)

論理結合子は，**論理演算子**(logical operator) と呼ばれることもある．

2.1.4 命題論理式の定義

定義 1 (命題論理式) 命題論理式の構文を以下のように再帰的に定義する．

(1) 命題定数および命題変数は命題論理式である．

(2) A, B が命題論理式のとき，以下も命題論理式である．

$$(\neg A)$$
$$(A \wedge B)$$
$$(A \vee B)$$
$$(A \Rightarrow B)$$

命題論理式の部分に現れる命題論理式を，**部分論理式**という．なお，命題論理式そのものも部分論理式に含めることにする．

練習問題 2.1.1 以下のうち命題論理式はどれか (上の定義に厳密に従って考えること)．

$$p,\ \neg p,\ (\neg p),\ \neg(p),\ (p),\ (p \wedge q \vee r),\ ((\neg p) \wedge (q \Rightarrow r))$$

練習問題 2.1.2 以下の命題論理式の部分論理式をすべて書き出せ．

$$(\neg q),\ ((\neg q) \Rightarrow r),\ ((\neg p) \wedge ((\neg q) \Rightarrow r))$$

- $(\neg A)$ は A の否定を表し，「A でない」,「not A」などと読む．$(\sim A)$ や $\overline{A}$ 等と記述する場合もある．
- $(A \wedge B)$ は A と B の連言を表し，「A かつ B」,「A and B」などと読む．$(A \& B)$ や $(A \cdot B)$ 等と記述する場合もある．
- $(A \vee B)$ は A と B の選言を表し，「A または B」,「A or B」などと読む．$(A + B)$ 等と記述する場合もある．

- $(A \Rightarrow B)$ は含意を表し，「A ならば B」，「A implies B」などと読む．$(A \supset B)$ や $(A \to B)$ 等と記述する場合もある．

上記の定義の通りだと，カッコが多く冗長なため，簡便のため以下のようにする．

- $\neg$ はできるだけ狭くかかるとする．すなわち $(\neg A \wedge B)$ は $((\neg A) \wedge B)$ を表す．
- $\wedge$, $\vee$, $\Rightarrow$ は右結合的とする．すなわち $(A \wedge B \wedge C)$ は $(A \wedge (B \wedge C))$ を表す．
- 一番外側のカッコは省略する．

練習問題 2.1.3　$\neg\neg p$ の省略されたカッコをすべて補って書け．

練習問題 2.1.4　$\neg p \wedge q \wedge \neg r$ の省略されたカッコをすべて補って書け．

練習問題 2.1.5　$(p \Rightarrow ((\neg q) \Rightarrow r))$ のカッコをできるだけ省略して書け．

練習問題 2.1.6　$(((\neg p) \wedge q) \wedge (p \vee q))$ のカッコをできるだけ省略して書け．

2.2　意味論 (semantics)

前節で構文を定義した命題論理式について，どのような場合に真の値となりどのような場合に偽の値となるのかを定める．

ここでは，真という値を 1 で表し，偽という値を 0 で表す．命題論理式の取り得る値は，1 か 0 かのどちらかであり，これらを命題論理式の**真理値**(truth value) という．

2.2.1　真理値表 (truth table)

命題論理式の真理値は，通常の数式の場合と同様に，部分論理式の真理値から定められる．

各論理結合子について，部分論理式の真理値からの計算方法は，以下のような**真理値表**にまとめることができる．

A	$\neg A$
0	1
1	0

A	B	$A \wedge B$
0	0	0
0	1	0
1	0	0
1	1	1

A	B	$A \vee B$
0	0	0
0	1	1
1	0	1
1	1	1

A	B	$A \Rightarrow B$
0	0	1
0	1	1
1	0	0
1	1	1

具体的な論理式の真理値は，その論理式に現れる命題変数の取り得る真理値の場合に応じて真理値表にまとめることができる．

例えば，$\neg p \vee q$ の真理値表は以下のようになる．

p	q	$\neg p$	$\neg p \vee q$
0	0	1	1
0	1	1	1
1	0	0	0
1	1	0	1

練習問題 2.2.1　$(\neg p \wedge q) \vee (p \wedge \neg q)$ の真理値表を書け．

練習問題 2.2.2　$(p \vee q) \wedge \neg(p \wedge q)$ の真理値表を書け．

練習問題 2.2.3 $p \Rightarrow q \Rightarrow p$ の真理値表を書け．

練習問題 2.2.4 n 個の異なる命題変数が現れている論理式の真理値表は，何行になるか．

2.2.2 付値 (assignment)

命題論理式の真理値は，その論理式に現れている各命題変数に対する真理値を定めることで，定めることができる．

各命題変数に対して真理値を定める関数を**付値**という．すなわち，今考えている命題変数の集合を V とした時，付値は V から $\{0,1\}$ への写像である．

付値 $v : V \to \{0,1\}$ は，以下を満たす V 中の命題変数を含む論理式上の関数として一意的に拡張できる．

$$\begin{aligned} v(\bot) &= 0 \\ v(\top) &= 1 \\ v(\neg A) = 1 &\iff v(A) = 0 \\ v(A \land B) = 1 &\iff v(A) = 1 \text{ かつ } v(B) = 1 \\ v(A \lor B) = 1 &\iff v(A) = 1 \text{ または } v(B) = 1 \\ v(A \Rightarrow B) = 1 &\iff v(A) = 0 \text{ または } v(B) = 1 \end{aligned}$$

あるいは，真理値 0 と 1 を自然数と解釈し，以下のように定義することもできる ($+$, $-$, $\cdot$ は自然数上の加減乗算である)．

$$\begin{aligned} v(\bot) &= 0 \\ v(\top) &= 1 \\ v(\neg A) &= 1 - v(A) \\ v(A \land B) &= v(A) \cdot v(B) \\ v(A \lor B) &= v(A) + v(B) - v(A) \cdot v(B) \\ v(A \Rightarrow B) &= 1 - v(A) + v(A) \cdot v(B) \end{aligned}$$

練習問題 2.2.5 付値 v が $v(p) = 0$, $v(q) = 0$ を満たすとき，$v(\neg p \lor q)$ の値は何か．

練習問題 2.2.6 付値 v が $v(p) = 1$, $v(q) = 0$ を満たすとき，$v(\neg p \lor q)$ の値は何か．

練習問題 2.2.7 上記の計算は，$\neg p \lor q$ の真理値表のどの行に対応しているか．

練習問題 2.2.8 命題変数の有限集合 V の要素の個数を $|V|$ で表した時，V 上の付値は何通りあるか．

2.2.3 恒真 (valid) と充足可能 (satisfiable)

$p \Rightarrow q \Rightarrow p$ は，すべての付値 v に対し $v(p \Rightarrow q \Rightarrow p) = 1$ となる．このような論理式は恒真な論理式と呼ばれる．

定義 2 (恒真 (valid)) 論理式 A が，すべての付値 v に対し $v(A) = 1$ となる時，A は**恒真**(valid) あるいは**トートロジー**(tautology) であるという．

$$\text{論理式 } A \text{ が恒真} \overset{\text{def}}{\iff} \text{すべての付値 } v \text{ について } v(A) = 1$$

定義 3 (充足可能 (satisfiable)) 論理式 A が，ある付値 v に対し $v(A) = 1$ となる時，A は**充足可能**(satisfiable) であるという．

$$\text{論理式 } A \text{ が充足可能} \overset{\text{def}}{\iff} \text{ある付値 } v \text{ について } v(A) = 1$$

論理式 A が充足可能でない時，A は**充足不能**(unsatisfiable) あるいは**恒偽**であるという．

明らかに，論理式 A が恒真ならば A は充足可能である．また，以下が成り立つ．

定理 1

$$\text{論理式 } \neg A \text{ が恒真} \quad \iff \text{論理式 } A \text{ が充足不能}$$

練習問題 2.2.9 以下のそれぞれの論理式は恒真か? 充足可能か?

$(p \wedge q) \Rightarrow (p \vee r)$, $p \wedge q$, $p \wedge \neg p$, $\neg p \vee p$

練習問題 2.2.10 上の定理「論理式 $\neg A$ が恒真 $\iff$ 論理式 A が充足不能」を証明せよ．

2.3 命題論理式のいくつかの性質

定義 4 (論理的同値 (logical equivalence)) 論理式 A, B が，すべての付値に対して真理値が一致するとき，A と B は**論理的に同値**であると言い $A \equiv B$ と書く．

$$A \equiv B \overset{\text{def}}{\iff} \text{すべての付値 } v \text{ について } v(A) = v(B)$$

ここで用いている $\equiv$ は論理結合子ではなく，$A \equiv B$ は論理式ではないことに注意する．

定理 2 (論理式の同値関係)

	$A \wedge \bot \equiv \bot$	$A \vee \top \equiv \top$
	$A \wedge \top \equiv A$	$A \vee \bot \equiv A$
べき等法則	$A \wedge A \equiv A$	$A \vee A \equiv A$
交換法則	$A \wedge B \equiv B \wedge A$	$A \vee B \equiv B \vee A$
結合法則	$A \wedge (B \wedge C) \equiv (A \wedge B) \wedge C$	$A \vee (B \vee C) \equiv (A \vee B) \vee C$
分配法則	$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$	$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$
二重否定の法則	$\neg\neg A \equiv A$	
矛盾律，排中律	$A \wedge \neg A \equiv \bot$	$A \vee \neg A \equiv \top$
ド・モルガンの法則	$\neg(A \wedge B) \equiv \neg A \vee \neg B$	$\neg(A \vee B) \equiv \neg A \wedge \neg B$
	$A \Rightarrow B \equiv \neg A \vee B$	
	$\neg A \equiv A \Rightarrow \bot$	
対偶	$A \Rightarrow B \equiv \neg B \Rightarrow \neg A$	

練習問題 2.3.1 真理値表を用いて，上の同値関係が成り立つことを示せ．

練習問題 2.3.2 $\bot$, $\top$, $\wedge$, $\vee$, $\neg$ のそれぞれが，空集合 $\emptyset$, 全体集合 U, 共通集合 $\cap$, 和集合 $\cup$, 補集合 $()^c$ を表していると考えて，上の同値関係を解釈してみよ．

練習問題 2.3.3 $\bot$, $\top$, $\wedge$, $\vee$, $\neg$ のそれぞれが，数値の 0 と 1, 最小値 (min), 最大値 (max), 演算 $1 - x$ を表していると考えて，上の同値関係を解釈してみよ．

C を命題論理式，p を命題変数とする (C 中に現れていなくても良い)．$C[A]$ により，C 中に現れているすべての p を A で置き換えた論理式を表す.

定理 3

$$A \equiv B \quad \Longrightarrow C[A] \equiv C[B]$$

(証明) 命題論理式の定義に従い帰納的に証明する (**論理式の構成に関する帰納法**)．すなわち，$A \equiv B$ の場合，論理式 C の任意の部分論理式について定理の結論が成立していると仮定し，どのような論理式についても結論が成立することを示す.

- C が命題定数の場合，$C[A]$ と $C[B]$ は同一であり $C[A] \equiv C[B]$ である.
- C が命題変数であり p に一致していない場合，$C[A]$ と $C[B]$ は同一であり $C[A] \equiv C[B]$ である.
- C が命題変数であり p と一致している場合，$C[A]$ は A に一致し，$C[B]$ は B に一致する．$A \equiv B$ より $C[A] \equiv C[B]$ である.
- C が $D \wedge E$ の形をしている場合，帰納法の仮定より $D[A] \equiv D[B]$ かつ $E[A] \equiv E[B]$，すなわち任意の付値 v について $v(D[A]) = v(D[B])$ かつ $v(E[A]) = v(E[B])$ である．したがって，任意の付値 v について以下が成り立つ.

$$\begin{aligned} v(C[A]) &= v(D[A] \wedge E[A]) \\ &= v(D[A]) \cdot v(E[A]) \\ &= v(D[B]) \cdot v(E[B]) \\ &= v(D[B] \wedge E[B]) \\ &= v(C[B]) \end{aligned}$$

  よって $C[A] \equiv C[B]$ である.
- C が $\neg D$, $D \vee E$, $D \Rightarrow E$ の形をしている場合も，上と同様にして $C[A] \equiv C[B]$ が示される.
- 論理式 C の構成の仕方は，以上ですべてであるので，任意の論理式 C について $C[A] \equiv C[B]$ である.

この定理により，基本的な同値関係を利用して，論理式を同値な別の論理式に変形することができる.

練習問題 2.3.4 上の同値関係を用いて論理式を変形し，$A \wedge B \equiv \neg(\neg A \vee \neg B)$ および $A \vee B \equiv \neg(\neg A \wedge \neg B)$ を示せ.

練習問題 2.3.5 上の同値関係を用いて論理式を変形し，$(\neg A \wedge B) \vee (A \wedge \neg B) \equiv (A \vee B) \wedge \neg(A \wedge B)$ を示せ.

練習問題 2.3.6 上の同値関係を用いて論理式を変形し，$A \Rightarrow (B \Rightarrow C) \equiv (A \wedge B) \Rightarrow C$ を示せ.

2.3.1 **双対性** (duality)

定義 5 (双対な論理式) 命題論理式 A は $\Rightarrow$ を含まないとする．A 中の $\bot$ を $\top$ で，$\top$ を $\bot$ で，$\wedge$ を $\vee$ で，$\vee$ を $\wedge$ で置き換えた論理式を A に**双対な論理式**という.

論理的に同値な論理式はその双対も論理的に同値になるという双対定理を証明する前に，以下の補題を示す.

補題 1 命題論理式 A は $\Rightarrow$ を含まないとし，A' は A と双対な論理式とする．v を付値とし，v' はすべての命題変数 p について $v'(p) = 1 - v(p)$ を満たす付値とする．この時，以下が成り立つ．

$$v(A) = 1 - v'(A')$$

(証明) 論理式 A の構成に関する帰納法を用い，$v(A) = 1 - v(A')$ を証明する．

- A が $\bot$ の場合．$v(A) = 0$, $v'(A') = v(\top) = 1$ であり証明すべき主張は成り立つ．
- A が $\top$ の場合．$v(A) = 1$, $v'(A') = v(\bot) = 0$ であり証明すべき主張は成り立つ．
- A が命題変数である場合．v' の定義より，$v(A) = 1 - v'(A)$ である．
- A が $B \wedge C$ の形をしている場合．A' は $B' \vee C'$ であり，帰納法の仮定より $v(B) = 1 - v'(B')$ かつ $v(C) = 1 - v'(C')$ である．

$$\begin{aligned}
v(A) &= v(B \wedge C) \\
&= v(B) \cdot v(C) \\
&= (1 - v'(B')) \cdot (1 - v'(C')) \\
&= 1 - (v'(B') + v'(C') - v'(B') \cdot v'(C')) \\
&= 1 - v'(B' \vee C') \\
&= 1 - v'(A')
\end{aligned}$$

- A が $B \vee C$ の形をしている場合．A' は $B' \wedge C'$ であり，帰納法の仮定より $v(B) = 1 - v'(B')$ かつ $v(C) = 1 - v'(C')$ である．

$$\begin{aligned}
v(A) &= v(B \vee C) \\
&= v(B) + v(C) - v(B) \cdot v(C) \\
&= (1 - v'(B')) + (1 - v'(C')) - (1 - v'(B')) \cdot (1 - v'(C')) \\
&= 1 - v'(B') \cdot v'(C')) \\
&= 1 - v'(B' \wedge C') \\
&= 1 - v'(A')
\end{aligned}$$

定理 4 (双対定理) 命題論理式 A, B は $\Rightarrow$ を含まないとし，A' と B' は，それぞれ A と B に双対な論理式とする．この時，以下が成り立つ．

$$A \equiv B \Longleftrightarrow A' \equiv B'$$

(証明)

($\Longrightarrow$) 論理的に同値の定義より，任意の付値 v について $v(A) = v(B)$ である．また，補題より $v(A) = 1 - v(A')$ かつ $v(B) = 1 - v(B')$ であるので，任意の付値 v について $v(A') = 1 - v(A) = 1 - v(B) = v(B')$，すなわち $A' \equiv B'$ である．

($\Longleftarrow$) A' および B' に双対な論理式は，それぞれ A および B と同一なので，$A \equiv B$ である．

2.3.2 他の論理結合子

これまで説明した NOT ($\neg$), AND ($\wedge$), OR ($\vee$), IMP ($\Rightarrow$) 以外でよく用いられる論理結合子には XOR ($\oplus$), NAND (not AND), NOR (not OR) がある．

A	B	$A \oplus B$
0	0	0
0	1	1
1	0	1
1	1	0

A	B	$\mathrm{NAND}(A, B)$
0	0	1
0	1	1
1	0	1
1	1	0

A	B	$\mathrm{NOR}(A, B)$
0	0	1
0	1	0
1	0	0
1	1	0

それぞれは，以下の同値関係を満たす．

$$
\begin{aligned}
A \oplus B &\equiv (\neg A \wedge B) \vee (A \wedge \neg B) \\
\mathrm{NAND}(A, B) &\equiv \neg(A \wedge B) \\
\mathrm{NOR}(A, B) &\equiv \neg(A \vee B)
\end{aligned}
$$

練習問題 2.3.7　真理値表を用いて，上の同値関係を確かめよ．

練習問題 2.3.8　$\neg A \equiv \mathrm{NAND}(A, \top)$ および $A \wedge B \equiv \mathrm{NAND}(\mathrm{NAND}(A, B), \top)$ を示せ．

特に，XOR ($\oplus$) は**排他的論理和**(Exclusive OR) と呼ばれ，以下のような性質がある．

	$A \oplus \bot \equiv A$
	$A \oplus \top \equiv \neg A$
	$A \oplus A \equiv \bot$
	$A \oplus \neg A \equiv \top$
交換法則	$A \oplus B \equiv B \oplus A$
結合法則	$A \oplus (B \oplus C) \equiv (A \oplus B) \oplus C$

練習問題 2.3.9　真理値表を用いて，上の同値関係を確かめよ．

練習問題 2.3.10　$A_1 \oplus A_2 \oplus \cdots \oplus A_n$ の真理値は，真理値が真となる A_i の個数の偶奇 (パリティ) に一致すること，すなわち $v(A_1 \oplus A_2 \oplus \cdots A_n) = (\sum_i v(A_i)) \bmod 2$ を示せ．

2.4　論理式の標準形

2.4.1　真理値表による方法

真理値表の節では，任意の命題論理式について，その真理値表を求める方法を学んだ．

では逆に，真理値表が与えられた時に，そのような真理値表を持つ論理式を求めることは可能だろうか．たとえば，以下のような真理値表を持つ論理式 X はどのようなものだろうか．

p	q	X
0	0	1
0	1	0
1	0	0
1	1	1

これは，以下の真理値表を眺めることで解決できる．

p	q	$\neg p \wedge \neg q$	$\neg p \wedge q$	$p \wedge \neg q$	$p \wedge q$
0	0	1	0	0	0
0	1	0	1	0	0
1	0	0	0	1	0
1	1	0	0	0	1

上の真理値表より，$\neg p \wedge \neg q$ は，p と q が共に偽の時のみ真になる論理式であることがわかる．また，$p \wedge q$ は p と q が共に真の時のみ真になる論理式であることがわかる．

したがって，X は p と q が共に偽の時および共に真の時のみ真になる論理式であるから，$(\neg p \wedge \neg q) \vee (p \wedge q)$ として表すことができる．

練習問題 2.4.1 以下の真理値表を持つ論理式 Y を，上の方法で求めよ．

p	q	Y
0	0	1
0	1	0
1	0	1
1	1	1

練習問題 2.4.2 以下の真理値表を持つ論理式 Z を，上と同様の方法で求めよ．この真理値表に対応する論理回路は，多数決回路と呼ばれる．

p	q	r	Z
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	0
1	0	1	1
1	1	0	1
1	1	1	1

練習問題 2.4.3 $(p \wedge q) \vee (q \wedge r) \vee (r \wedge p)$ の真理値表を書け．

また，同様に以下の真理値表を用い，上の方法と双対的に偽の場合に着目すれば，X を $(\neg p \vee q) \wedge (p \vee \neg q)$ として表すことができる．

p	q	$p \vee q$	$p \vee \neg q$	$\neg p \vee q$	$\neg p \vee \neg q$
0	0	0	1	1	1
0	1	1	0	1	1
1	0	1	1	0	1
1	1	1	1	1	0

一つ目の方法では, 命題変数あるいは命題変数の否定のいくつかを連言 ($\wedge$) で結び, それらをさらに選言 ($\vee$) で結んだ式が得られている ($(\neg p \wedge \neg q) \vee (p \wedge q)$)．このような形の論理式は，**選言標準形**(**DNF**, Disjunctive Normal Form) あるいは**加法標準形**，**積和標準形**と呼ばれる．

二つ目の方法では, 命題変数あるいは命題変数の否定のいくつかを選言 ($\vee$) で結び, それらをさらに連言 ($\wedge$) で結んだ式が得られている ($(\neg p \vee q) \wedge (p \vee \neg q)$)．このような形の論理式は，**連言標準形**(**CNF**, Conjunctive Normal Form) あるいは**乗法標準形**，**和積標準形**と呼ばれる．

定義 6 (選言標準形と連言標準形) 命題定数，命題変数あるいは命題変数の否定を**リテラル**(literal) と呼ぶ．一つ以上のリテラルの連言を**基本積**，一つ以上のリテラルの選言を**基本和**と呼ぶ．一つ以上の基本積の選言を**選言標準形**(DNF)，一つ以上の基本和の連言を**連言標準形**(CNF) と呼ぶ．

一般に，n 個の真理値を入力として一つの真理値を定める関数を，n 変数の**論理関数**という．これまでの議論から，以下のことがわかる．

定理 5 任意の n 変数論理関数は，n 個の命題変数を含む選言標準形および連言標準形の命題論理式として表現できる．したがって，任意の命題論理式は，それと論理的に同値な選言標準形および連言標準形の命題論理式として表現できる．

一つの論理式が，選言標準形でもあり連言標準形でもある場合があることに注意する．たとえば $p \vee q$ は選言標準形でもあり連言標準形でもある．

また，ある論理式と同値な標準形は一つに限らないことにも注意する．$p \vee q$ は，それ自体が選言標準形であるが，$p \vee q \equiv (\neg p \wedge q) \vee (p \wedge \neg q) \vee (p \wedge q)$ であり，右辺は選言標準形である．

2.4.2 式の変形による方法

上では，真理値表を用いて標準形を求めたが，式の変形によって求めることもできる．

そのために，以下の手順に従い，まず**否定標準形**(NNF, Negation Normal Form) を求める．

(1) 以下の同値関係を用い，左辺を右辺の式で置き換えることによって，元の論理式から $\Rightarrow$ を削除する．

$$A \Rightarrow B \equiv \neg A \vee B$$

(2) 以下の同値関係を繰り返し用い，否定が命題変数の直前にのみ現れるようにする．

$$\begin{aligned} \neg\bot &\equiv \top \\ \neg\top &\equiv \bot \\ \neg\neg A &\equiv A \\ \neg(A \wedge B) &\equiv \neg A \vee \neg B \\ \neg(A \vee B) &\equiv \neg A \wedge \neg B \end{aligned}$$

否定標準形に対して，以下の分配法則 (および交換法則と結合法則) を繰り返し用いることで，選言標準形を求めることができる．

$$A \wedge (B \vee C) \equiv (A \wedge B) \vee (A \wedge C)$$

同様に，否定標準形に対して，以下の分配法則 (および交換法則と結合法則) を繰り返し用いることで，連言標準形を求めることができる．

$$A \vee (B \wedge C) \equiv (A \vee B) \wedge (A \vee C)$$

なお，必要に応じて以下を用いることで，より簡潔な標準形を求めることができる．

$$\begin{array}{rcl@{\qquad}rcl} A \wedge \bot & \equiv & \bot & A \vee \top & \equiv & \top \\ A \wedge \top & \equiv & A & A \vee \bot & \equiv & A \\ A \wedge \neg A & \equiv & \bot & A \vee \neg A & \equiv & \top \end{array}$$

練習問題 2.4.4 $p \Rightarrow q$ の選言標準形，連言標準形を求めよ．

練習問題 2.4.5 $(p \vee q) \wedge \neg(p \wedge q)$ の選言標準形，連言標準形を求めよ．

練習問題 2.4.6 任意の命題論理式は，論理結合子として $\neg$ と $\wedge$ のみを含む同値な命題論理式として表現できることを示せ．(ヒント: $\vee$ を $\neg$ と $\wedge$ で表すことを考える)

練習問題 2.4.7 任意の命題論理式は，論理結合子として $\neg$ と $\vee$ のみを含む同値な命題論理式として表現できることを示せ．(ヒント: $\wedge$ を $\neg$ と $\vee$ で表すことを考える)

練習問題 2.4.8　任意の命題論理式は，論理結合子として $\Rightarrow$ のみを含む同値な命題論理式として表現できることを示せ．（ヒント: $\neg A \equiv A \Rightarrow \bot$ を用いる）

練習問題 2.4.9　任意の命題論理式は，論理結合子として NAND のみを含む同値な命題論理式として表現できることを示せ．

練習問題 2.4.10　任意の命題論理式は，論理結合子として NOR のみを含む同値な命題論理式として表現できることを示せ．

2.5 推論

ここでは，いくつかの命題論理式から別の新たな命題論理式を導き出す，すなわち別の新たな命題論理式を**推論**(inference) する方法について学ぶ．

命題論理について，形式的 (記号的) に推論を行うための**推論体系**(inference system) には，以下のように様々な種類がある．

- 導出原理
- ヒルベルトの体系
- 自然演繹 (ゲンツェンの体系 NK)
- シーケント計算 (ゲンツェンの体系 LK)

これらは，いずれも真と仮定する命題論理式の集合 Δ から，新しい命題論理式 A を論理的に推論するための体系である．これらの体系には，真と仮定している命題論理式やすでに導き出された命題論理式から，新たな命題を導き出すための**推論規則**(inference rule) が備わっている．

一般に，推論体系で前提 Δ から何回かの推論規則を適用して A が導き出された時，以下のように書く．

$$\Delta \vdash A$$

2.6 命題論理の節形式と導出原理

2.6.1 節形式 (clausal form)

- リテラル: 命題変数またはその否定 (例: p, $\neg p$, q, $\neg q$)
- 節 (clause): 0 個以上のリテラルの集合．リテラルの選言を表す．(例: $\{p, \neg q\}$, $\{\neg p, q\}$)
- 節集合: 0 個以上の節の集合．節の連言を表す．(例: $\{\{p, \neg q\}, \{\neg p, q\}\}$)

命題論理の節形式は，乗法標準形のことである．

空集合の節は空節と呼ばれ，$\Box$ で表される．空節は矛盾 ($\bot$) を意味する．

一般に，節 $\{\neg p_1, \neg p_2, \ldots, \neg p_m, q_1, q_2, \ldots, q_n\}$ は $(p_1 \wedge p_2 \wedge \cdots \wedge p_m) \Rightarrow (q_1 \wedge q_2 \wedge \cdots \wedge q_n)$ と同値である．

2.6.2 節形式への変換方法

(1) $A \Rightarrow B$ の形の論理式を $\neg A \vee B$ に変換する．

(2) ド・モルガンの法則および二重否定の法則を用いて，リテラル以外に $\neg$ が現れないように変換する．

(3) 分配法則を用いて，乗法標準形に変換する．

練習問題 2.6.1 $((p \Rightarrow q) \wedge (q \Rightarrow r)) \wedge \neg(p \Rightarrow r)$ を節形式に変換せよ．

練習問題 2.6.2 $(p \Rightarrow ((q \Rightarrow \neg r) \wedge \neg(r \Rightarrow \neg q))) \wedge p$ を節形式に変換せよ．

2.6.3 導出 (resolution)

$C_1 = \{P_1, \ldots, P_m\}, C_2 = \{Q_1, \ldots, Q_n\}$ を節とする．また，P_1 が命題変数で Q_1 がその否定となっているとする (すなわち $\neg P_1 = Q_1$)．このとき節 $C = \{P_2, \ldots, P_m, Q_2, \ldots, Q_n\}$ を得ることを導出と呼び，C を導出節と呼ぶ．C はリテラルの集合であるので，同一のリテラルは一つにまとめられる点に注意すること．

2.6.4 導出原理 (resolution principle)

$$
\begin{aligned}
& \text{論理式 } A \Rightarrow B \text{ は恒真} \\
\Longleftrightarrow\ & \text{論理式 } A \wedge \neg B \text{ は充足不能} \\
\Longleftrightarrow\ & \text{論理式 } A \wedge \neg B \text{ の節形式から空節が導出可能}
\end{aligned}
$$

導出原理は健全かつ完全である．

練習問題 2.6.3 節集合 $\{\{\neg p, q, r\},\ \{\neg q, s\},\ \{\neg q, s\}\}$ から節 $\{\neg p, s\}$ を導出せよ．

練習問題 2.6.4 節集合 $\{\{\neg p, \neg q\},\ \{\neg p, q\},\ \{p, \neg q\},\ \{p, q\}\}$ から空節を導出せよ．

練習問題 2.6.5 $((p \Rightarrow q) \wedge (q \Rightarrow r)) \wedge \neg(p \Rightarrow r)$ の節形式から空節を導出せよ．

練習問題 2.6.6 $(p \Rightarrow ((q \Rightarrow \neg r) \wedge \neg(r \Rightarrow \neg q))) \wedge p$ の節形式から空節を導出せよ．

2.7 SAT 問題と SAT ソルバー

2.7.1 SAT 問題

- 与えられた節形式の命題論理式が充足可能 (SAT) か充足不能 (UNSAT) かを判定する問題
- 充足可能の場合は，論理式を真にする付値のひとつを求める
- SAT 問題は決定可能 (decidable) な問題
- SAT 問題は初めて NP 完全であることが証明された問題

2.7.2 SAT ソルバー

- SAT 問題を解くプログラム
- 大きく二種類に分類される
  - Complete (Systematic，系統的) SAT Solver
    - SAT/UNSAT を答える
    - DPLL アルゴリズムに基づくものが主流
    - 2000 年頃から非常に高速なソルバーが開発されている
  - Incomplete (Stochastic，確率的) SAT Solver
    - SAT のみを答える

- SAT Solver Competetion
  - 2002 年からほぼ毎年開催
  - ソルバーのプログラムが公開されている
  - 最大数千万リテラルの規模のベンチマーク問題

2.7.3 SAT **変換** (SAT encoding)

- 種々の探索問題を SAT 問題に変換して求解する手法
- 近年，応用が盛ん
  - プランニング問題
  - スケジューリング問題
  - ハードウェア・ソフトウェアの検証
  - 制約充足問題
- SAT 問題を探索問題のアセンブリ言語としてとらえる

2.7.4 SAT **ソルバーの例**

- MiniSat
  - Complete SAT Solver
  - 非常に高速
  - 非常に小さい (2 千〜3 千行)
- 入力例 (DIMACS 形式)

`p cnf 3 4`	# 3 変数 4 節
`1 2 3 0`	# $p_1 \vee p_2 \vee p_3$
`-1 -2 0`	# $\neg p_1 \vee \neg p_2$
`1 -3 0`	# $\neg p_1 \vee \neg p_3$
`-2 -3 0`	# $\neg p_2 \vee \neg p_3$

- 出力例

`SAT`	# 充足可能
`-1 2 -3 0`	# $p_1 = 0, p_2 = 1, p_3 = 0$

2.7.5 **グラフ彩色問題** (GCP) **の** SAT **変換の例**

グラフ彩色問題 (GCP, Graph Coloring Problem)

```
p edge 4 5
e 1 2
e 1 3
e 2 3
e 2 4
e 3 4
```

3 色で彩色可能かの制約充足問題 (CSP, Constraint Satisfaction Problem)

$$
\begin{gathered}
x_1 \in \{1,2,3\} \\
x_2 \in \{1,2,3\} \\
x_3 \in \{1,2,3\} \\
x_4 \in \{1,2,3\} \\
x_1 \neq x_2 \\
x_1 \neq x_3 \\
x_2 \neq x_3 \\
x_2 \neq x_4 \\
x_3 \neq x_4
\end{gathered}
$$

Direct encoding により，SAT 問題に変換する．各整数変数 x_i $(i = 1,2,3,4)$ およびそれらの取り得る値 a $(a = 1,2,3)$ に応じて，12 個の命題変数 $p_{i,a}$ を用意する．

$$p_{i,a} \Longleftrightarrow x_i = a$$

各整数変数 x_i について，x_i が 1 か 2 か 3 かのいずれかの値を一つだけ取るという意味の以下の節を用意する．

$$
\begin{gathered}
p_{i,1} \vee p_{i,2} \vee p_{i,3} \\
\neg p_{i,1} \vee \neg p_{i,2} \\
\neg p_{i,2} \vee \neg p_{i,3} \\
\neg p_{i,3} \vee \neg p_{i,1}
\end{gathered}
$$

各制約条件 $x_i \neq x_j$ に対して，x_i と x_j の値が異なるという意味の以下の節を用意する．

$$
\begin{gathered}
\neg p_{i,1} \vee \neg p_{j,1} \\
\neg p_{i,2} \vee \neg p_{j,2} \\
\neg p_{i,3} \vee \neg p_{j,3}
\end{gathered}
$$

すべての節を DIMACS 形式で記述すると以下のような SAT 問題になる．

```
p cnf 12 31
1 2 3 0
-1 -2 0
-1 -3 0
-2 -3 0
4 5 6 0
-4 -5 0
-4 -6 0
-5 -6 0
7 8 9 0
-7 -8 0
-7 -9 0
-8 -9 0
10 11 12 0
-10 -11 0
-10 -12 0
-11 -12 0
-1 -4 0
-2 -5 0
-3 -6 0
```

```
-1 -7 0
-2 -8 0
-3 -9 0
-4 -7 0
-5 -8 0
-6 -9 0
-4 -10 0
-5 -11 0
-6 -12 0
-7 -10 0
-8 -11 0
-9 -12 0
```

これを SAT ソルバーに解かせると，以下の結果が得られる．

```
SAT
-1 -2 3 4 -5 -6 -7 8 -9 -10 -11 12 0
```

これは，以下の結果を表している．

$$x_1 = 3,\ x_2 = 1,\ x_3 = 2,\ x_4 = 3$$

2.7.6 Tseitin 変換

練習問題 2.7.1 命題変数 $p_1, p_2, \ldots, p_4$ のうち，1 個だけが真である時かつその時に限り真となる論理式を書け．

練習問題 2.7.2 命題変数 $p_1, p_2, \ldots, p_4$ のうち，ちょうど 2 個が真である時かつその時に限り真となる論理式を書け．

練習問題 2.7.3 命題変数 $p_1, p_2, \ldots, p_4$ のうち，2 個以上が真である時かつその時に限り真となる論理式を書け．

練習問題 2.7.4 美術館パズルを SAT 問題に変換して解く方法について考えよ．

練習問題 2.7.5 お絵描きロジック (ノノグラム，ピクロス) を SAT 問題に変換して解く方法について考えよ．

3 述語論理 (predicate logic)

- 個体定数: a, b, c など
- 個体変数: x, y, z など
- 関数記号: f, g, h など
- 述語記号: P, Q, R など

3.0.7 項の (再帰的) 定義

(1) 個体定数および個体変数は項である．

(2) $t_1, \ldots t_n$ が項で，f が関数記号のとき，$f(t_1, \ldots, t_n)$ も項である．

3.0.8 述語論理式の (再帰的) 定義

(1) 命題定数は述語論理式である．

(2) $t_1, \ldots t_n$ が項で，P が述語記号のとき，$P(t_1, \ldots, t_n)$ は述語論理式 (素式，原子論理式) である．

(3) A, B が述語論理式のとき，以下も述語論理式である．

$$(\neg A), \quad (A \wedge B), \quad (A \vee B), \quad (A \Rightarrow B)$$

(4) A が述語論理式で，x が個体変数のとき，以下も述語論理式である．

$$\begin{array}{ll} (\forall x A) & (\text{全称, for all, any: すべての } x \text{ について } A) \\ (\exists x A) & (\text{存在, exists, some: ある } x \text{ について } A) \end{array}$$

対象領域が有限なら，全称は連言で，存在は選言で表現できる．たとえば，対象領域 $D = \{a_1, \ldots, a_n\}$ とすると以下のようになる．

$$\forall x P(x) = P(a_1) \wedge \cdots \wedge P(a_n)$$
$$\exists x P(x) = P(a_1) \vee \cdots \vee P(a_n)$$

3.1 述語論理式のいくつかの性質

$$\begin{array}{cc} \forall x(A \wedge B) \equiv \forall x A \wedge \forall x B & \exists x(A \vee B) \equiv \exists x A \vee \exists x B \\ \neg \forall x A \equiv \exists x \neg A & \neg \exists x A \equiv \forall x \neg A \\ \neg \forall x(A \Rightarrow B) \equiv \exists x(A \wedge \neg B) & \neg \exists x(A \wedge B) \equiv \forall x(A \Rightarrow \neg B) \end{array}$$

練習問題 3.1.1 論理式による記述の練習 (別紙)

3.2 述語論理の節形式と導出原理

3.2.1 節形式 (clausal form)

- リテラル: 素式 (原子論理式) またはその否定．
- 節 (clause): 0 個以上のリテラルの集合．リテラルの選言の全称閉包を表す．(例: $\{P(x), \neg Q(x, y)\}$ は $\forall x \forall y(P(x) \vee \neg Q(x, y))$ を表す)
- 節集合: 0 個以上の節の集合．節の連言を表す．

3.2.2 節形式への変換方法

(1) $A \Rightarrow B$ の形の論理式を $\neg A \vee B$ に変換する．

(2) ド・モルガンの法則および二重否定の法則を用いて，リテラル以外に $\neg$ が現れないように変換する．

(3) $\exists y$ があれば，その外側に現れている全称束縛されている変数を $x_1, \ldots, x_n$ として，y を $f(x_1, \ldots, x_n)$ (スコーレム関数) で置き換える．ただし f は論理式中に現れていない新しい関数記号である．

(4) すべての変数名が異なるように，変数名を付け換える．

(5) 全称記号，存在記号を一番前に移す (冠頭形)．

(6) 分配法則を用いて，連言標準形に変換する．

3.2.3 導出

C_1, C_2 を節とし, それぞれの変数に適当な項を代入すると $\{P_1, \ldots, P_m\}, \{Q_1, \ldots, Q_n\}$ となり, P_1 が原子論理式で Q_1 がその否定となっているとする (すなわち $\neg P_1 = Q_1$). このとき節 $C = \{P_2, \ldots, P_m, Q_2, \ldots, Q_n\}$ を得ることを導出と呼び, C を導出節と呼ぶ. C はリテラルの集合であるので, 同一のリテラルは一つにまとめられる点に注意すること.

述語論理に対しても, 導出原理は健全かつ完全である.

練習問題 3.2.1 以下を導出法で証明せよ (Kowalski 著「論理による問題の解法」より).
すべてのキノコ (Fungus) は食用キノコ (Mushroom) であるか毒キノコ (Toadstool) である. すべてのイグチ属キノコ (Boletus) はキノコである. すべての毒キノコは毒である. イグチ属キノコは食用キノコではない. このとき, すべてのイグチ属キノコは毒であることを示せ.

練習問題 3.2.2 以下を導出法で証明せよ (Genesereth, Nilsson 著「人工知能基礎論」より).
もしも講座がやさしいものであれば, 幸せな受講生が存在する. もしも講座に期末試験があるのであれば, すべての受講生は不幸である. このとき, 期末試験のある講座はやさしくないことを示せ.

4 シーケント計算 (sequent calculus)

$$\frac{}{\Gamma_1, A, \Gamma_2 \longrightarrow \Delta_1, A, \Delta_2}\ (\mathrm{Ax})$$

$$\frac{\Gamma_1, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, \top, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\top) \qquad \frac{}{\Gamma \longrightarrow \Delta_1, \top, \Delta_2}\ (\mathrm{R}\top)$$

$$\frac{}{\Gamma_1, \bot, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\bot) \qquad \frac{\Gamma \longrightarrow \Delta_1, \Delta_2}{\Gamma \longrightarrow \Delta_1, \bot, \Delta_2}\ (\mathrm{R}\bot)$$

$$\frac{\Gamma_1, \Gamma_2 \longrightarrow A, \Delta}{\Gamma_1, \neg A, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\neg) \qquad \frac{A, \Gamma \longrightarrow \Delta_1, \Delta_2}{\Gamma \longrightarrow \Delta_1, \neg A, \Delta_2}\ (\mathrm{R}\neg)$$

$$\frac{\Gamma_1, A, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, A \wedge B, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\wedge) \qquad \frac{\Gamma \longrightarrow \Delta_1, A, \Delta_2 \quad \Gamma \longrightarrow \Delta_1, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, A \wedge B, \Delta_2}\ (\mathrm{R}\wedge)$$

$$\frac{\Gamma_1, A, \Gamma_2 \longrightarrow \Delta \quad \Gamma_1, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, A \vee B, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\vee) \qquad \frac{\Gamma \longrightarrow \Delta_1, A, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, A \vee B, \Delta_2}\ (\mathrm{R}\vee)$$

$$\frac{\Gamma_1, \Gamma_2 \longrightarrow A, \Delta \quad \Gamma_1, B, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, A \Rightarrow B, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L} \Rightarrow) \qquad \frac{A, \Gamma \longrightarrow \Delta_1, B, \Delta_2}{\Gamma \longrightarrow \Delta_1, A \Rightarrow B, \Delta_2}\ (\mathrm{R} \Rightarrow)$$

$$\frac{\Gamma_1, A[t/x], \forall x.A, \Gamma_2 \longrightarrow \Delta}{\Gamma_1, \forall x.A, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\forall) \qquad \frac{\Gamma \longrightarrow \Delta_1, A[y/x], \Delta_2}{\Gamma \longrightarrow \Delta_1, \forall x.A, \Delta_2}\ (\mathrm{R}\forall)$$

$$\frac{\Gamma_1, A[y/x], \Gamma_2 \longrightarrow \Delta}{\Gamma_1, \exists x.A, \Gamma_2 \longrightarrow \Delta}\ (\mathrm{L}\exists) \qquad \frac{\Gamma \longrightarrow \Delta_1, A[t/x], \exists x.A, \Delta_2}{\Gamma \longrightarrow \Delta_1, \exists x.A, \Delta_2}\ (\mathrm{R}\exists)$$

Here, y is not free in the lower sequent in (R$\forall$) and (L$\exists$).