

数理論理学 補助資料集

記号操作による推論の例	2
記号の優先度	3
記号による式の組み立て	4
真理表	5
プログラムによる真偽の計算	6
論理式の 略記と慣用表現	8
論理と集合 の基本法則	9
全称と存在 の否定	11
全称と存在 の併用	12
式の 再帰構造と木構造	14
式の帰納的定義	15
自然演繹における 仮定の有効範囲 と 変数条件	16
構造による式の解釈	17

山田 俊行

<https://www.cs.info.mie-u.ac.jp/~toshi/lectures/mathlogic/>

2024年4月

記号操作による推論の例

数理論理学では、証明を記号操作による推論として表して、その性質を数学的に論じる。

●記号操作による推論の例 (同値変形)

	【式変形に使う論理法則】
(1) $\neg \forall x (P(x) \Rightarrow Q(x))$	$A \Rightarrow B$ と $\neg A \vee B$ は同じ
↓	
(2) $\neg \forall x (\neg P(x) \vee Q(x))$	$\neg \forall x A$ と $\exists x \neg A$ は同じ
↓	
(3) $\exists x \neg (\neg P(x) \vee Q(x))$	$\neg (A \vee B)$ と $\neg A \wedge \neg B$ は同じ
↓	
(4) $\exists x (\neg \neg P(x) \wedge \neg Q(x))$	$\neg \neg A$ と A は同じ
↓	
(5) $\exists x (P(x) \wedge \neg Q(x))$	
	【記号の意味】
	$\neg A$ A ではない
	$A \wedge B$ A かつ B
	$A \vee B$ A または B
	$A \Rightarrow B$ A ならば B
	$\forall x A$ 全ての x について A
	$\exists x A$ ある x について A

式変形から分かること：命題 (1)~(5) は全て同じ意味 (真偽がいつでも一致)

式 (1) と (5) の文章表現：

(1') 「全ての x について、 x が P を満たすなら Q も満たす」とはいえない

(5') ある x について、 x が P を満たし、かつ、 Q を満たさない

より簡潔な文章表現：

(1'') 「全ての x について、 P ならば Q 」とはいえない

(5'') P であり Q でない対象 (反例) がある

●記号操作の応用例 (同値変形による読み替え)

集合の包含関係の否定 $S \not\subseteq T$ は、包含関係の定義より $\neg \forall x (x \in S \Rightarrow x \in T)$ 、つまり「 S に属すどの要素も T に属す、とはいえない」

上の同値変形を使って、より読みやすい式に変えたものは、 $\exists x (x \in S \wedge x \notin T)$ 、つまり「 S に属して T に属さない要素がある」

●記号で表す利点と欠点

利点：記号操作 (計算) だけで、正しく推論できる

欠点：記号の意味を知らないと、意図が分からない

記号の優先度

記号の優先度を定めれば、括弧を省いて式を読み易くできる。

●優先度の高さ

論理記号	実数に使う記号
高	高
1 \neg	1 \sin
2 $\wedge \vee$	2 $\times \div$
3 \Rightarrow	3 $+$ $-$
4 \Leftrightarrow	4 $=$
低	低

●括弧を省く例

優先度が高い記号を囲む括弧と、一番外側の括弧を省ける

$$\begin{array}{ccc} \left(\left(P \wedge (\neg Q) \right) \Rightarrow \left(\neg (Q \Rightarrow R) \right) \right) & & \left(\left(x \times (\sin y) \right) - \left(\sin (y - z) \right) \right) \\ \downarrow & & \downarrow \\ P \wedge \neg Q \Rightarrow \neg (Q \Rightarrow R) & & x \times \sin y - \sin (y - z) \end{array}$$

優先度が同じ記号を囲む括弧を省くこともある (結合法則を満たすとき)

$$\begin{array}{ccc} P \wedge (Q \wedge R) & & x \times (y \times z) \\ \downarrow & & \downarrow \\ P \wedge Q \wedge R & & x \times y \times z \end{array}$$

記号による式の組み立て

式を作るとき、記号ごとに使える式の種類が決まっている、という点に注意する。式の種類の違いを区別するため、以下の説明では、対象を $(\)$ で、述語を $[\]$ で囲って示す。

●式の正しい構成

関数記号	対象	→	対象	
	$(\)$		$(\)$	
+	$x \quad 2$ $(x) \quad (2)$		$x + 2$ $((x) + (2))$	x と 2 の和
\cap	$\mathbb{N} \quad A$ $(\mathbb{N}) \quad (A)$		$\mathbb{N} \cap A$ $((\mathbb{N}) \cap (A))$	\mathbb{N} と A の共通部分
述語記号	対象	→	述語	
	$(\)$		$[\]$	
	$y \quad 6$ $(y) \quad (6)$		$y \mid 6$ $[(y) \mid (6)]$	y は 6 の約数である
=	$z \quad 2$ $(z) \quad (2)$		$z = 2$ $[(z) = (2)]$	z は 2 に等しい
\subseteq	$\mathbb{N} \quad A$ $(\mathbb{N}) \quad (A)$		$\mathbb{N} \subseteq A$ $[(\mathbb{N}) \subseteq (A)]$	\mathbb{N} は A の部分集合である
論理結合子	述語	→	述語	
	$[\]$		$[\]$	
\wedge	$y \mid 6 \quad z = 2$ $[y \mid 6] \quad [z = 2]$		$y \mid 6 \wedge z = 2$ $[[y \mid 6] \wedge [z = 2]]$	$y \mid 6$ かつ $z = 2$

●式の誤った構成

「 2 と 3 は x の約数である」という主張の、誤った記号表現の例を二つ示す。

$$(2 \mid x) \cap (3 \mid x) \quad \text{関数記号 } \cap \text{ は述語に使えない}$$

$$(2 \wedge 3) \mid x \quad \text{論理結合子 } \wedge \text{ は対象に使えない}$$

対象と述語を括弧で囲んで、誤りの原因を分析するとよい。

限られた式にだけ使える記号も多いので、扱う対象の種類にも注意する。

$$2 \cap x \quad \text{集合演算 } \cap \text{ は集合以外の対象に使えない}$$

$$2 \subseteq x \quad \text{包含関係 } \subseteq \text{ は集合以外の対象に使えない}$$

真理表

複合命題の真理表を表示する C プログラムを紹介する。真理表の演習問題の答え合わせに使うとよい。

●プログラム

```
#include <stdio.h>

int main(void) {
    int a, b, c;    // 基本命題の真理値

    // 基本命題の真偽の組み合わせ全てに対して、複合命題の真理値を表示

    for (a = 0; a <= 1; a++)
        for (b = 0; b <= 1; b++)
            for (c = 0; c <= 1; c++)
                printf("%d %d %d | %d\n", a, b, c, !(a <= (b && c)));

    return 0;
}
```

このプログラムは、命題が $\neg(A \Rightarrow (B \wedge C))$ の場合なので、必要に応じて次の演算子を使う。記号の優先度の違いにも要注意。

! (否定) && (論理積) || (論理和) <= (含意) == (同値)

●実行結果

```
0 0 0 | 0
0 0 1 | 0
0 1 0 | 0
0 1 1 | 0
1 0 0 | 1
1 0 1 | 1
1 1 0 | 1
1 1 1 | 0
```

●考察課題

- (1) 論理演算として <= や == を使えるのはなぜか。
- (2) 命題を自由に指定できるようプログラムを変更せよ。

プログラムによる真偽の計算

全称命題や存在命題の真偽を計算する C プログラムの例を紹介する。

●プログラム

対象領域 (変数の動く範囲) が有限集合のとき、各対象 x についての述語 $P(x)$ の真偽が計算できれば、全称命題 $\forall x P(x)$ や存在命題 $\exists x P(x)$ の真偽を計算できる。

```
#include <stdio.h>

enum { MIN = -3, MAX = 3 }; // 対象領域 { -3, -2, 1, 0, 1, 2, 3 }

// 全称命題  $\forall x P(x)$  の真偽を  $P(\text{MIN}) \wedge \dots \wedge P(\text{MAX})$  の真偽として計算
int all(int P(int), char *str) {
    int truth = 1;
    int x;
    for (x = MIN; x <= MAX; x++) {
        int b = P(x); // P(x) の真偽
        truth = truth && b; // P(MIN)  $\wedge$   $\dots$   $\wedge$  P(x) の真偽
        printf("%s(%d) : %d\n", str, x, b);
    }
    printf("ALL x %s(x) : %d\n\n", str, truth);
    return truth; //  $\forall x P(x)$  の真偽
}

// 存在命題  $\exists x P(x)$  の真偽を  $P(\text{MIN}) \vee \dots \vee P(\text{MAX})$  の真偽として計算
int exist(int P(int), char *str) {
    int truth = 0;
    int x;
    for (x = MIN; x <= MAX; x++) {
        int b = P(x); // P(x) の真偽
        truth = truth || b; // P(MIN)  $\vee$   $\dots$   $\vee$  P(x) の真偽
        printf("%s(%d) : %d\n", str, x, b);
    }
    printf("EXIST x %s(x) : %d\n\n", str, truth);
    return truth; //  $\exists x P(x)$  の真偽
}

// 各述語の定義
int P1(int x) { return ((x>=0) <= (x>1)); } //  $x \geq 0 \Rightarrow x > 1$ 
int P2(int x) { return ((x>1) <= (x>=0)); } //  $x > 1 \Rightarrow x \geq 0$ 
int P3(int x) { return ((x%4==0) && !(x%2==0)); } //  $4|x \wedge \neg 2|x$ 
int P4(int x) { return (!(x%4==0) && (x%2==0)); } //  $\neg 4|x \wedge 2|x$ 

int main(void) {
    all(P1, "P1"); // 0 以上の数はみな 1 を超える  $\forall x (x \geq 0 \Rightarrow x > 1)$ 
    all(P2, "P2"); // 1 を超える数はみな 0 以上  $\forall x (x > 1 \Rightarrow x \geq 0)$ 
    exist(P3, "P3"); // 4 で割り切れる非偶数がある  $\exists x (4|x \wedge \neg 2|x)$ 
    exist(P4, "P4"); // 4 で割り切れない偶数がある  $\exists x (\neg 4|x \wedge 2|x)$ 
    return 0;
}
```

全称記号や存在記号を伴う論理式の真偽は、対象領域全体を動く変数を使った反復計算により求まる (上のプログラムでは、変数 x が MIN から MAX までを動く)。その他の論理記号の C プログラムでの扱い方については、p. 5 「真理表」の補助資料に解説がある。

●実行結果

前掲のプログラムの実行結果は次の通り.

P1(-3) : 1
P1(-2) : 1
P1(-1) : 1
P1(0) : 0
P1(1) : 0
P1(2) : 1
P1(3) : 1
ALL x P1(x) : 0

P2(-3) : 1
P2(-2) : 1
P2(-1) : 1
P2(0) : 1
P2(1) : 1
P2(2) : 1
P2(3) : 1
ALL x P2(x) : 1

P3(-3) : 0
P3(-2) : 0
P3(-1) : 0
P3(0) : 0
P3(1) : 0
P3(2) : 0
P3(3) : 0
EXIST x P3(x) : 0

P4(-3) : 0
P4(-2) : 1
P4(-1) : 0
P4(0) : 0
P4(1) : 0
P4(2) : 1
P4(3) : 0
EXIST x P4(x) : 1

上から順に, (1) $\forall x (x \geq 0 \Rightarrow x > 1)$, (2) $\forall x (x > 1 \Rightarrow x \geq 0)$, (3) $\exists x (4 \mid x \wedge \neg 2 \mid x)$, (4) $\exists x (\neg 4 \mid x \wedge 2 \mid x)$, の四つの論理式の真偽を計算している.

●考察課題

- (1) 全称命題 $\forall x P(x)$ と存在命題 $\exists x P(x)$ の真偽の定め方をふまえて, 上記の結果を説明せよ.
- (2) この資料には現れない述語を使って全称命題や存在命題を作り, 真偽を手計算で求めてからプログラムによりその正しさを確かめよ.
- (3) 2変数以上の述語と複数の量化子を使う論理式の例について, 真偽を計算するプログラムを作成せよ.

論理式の 略記と慣用表現

論理式の 略記・慣用表現 を知っておくと、数学的な主張を論理式で表すときに役立つ。
略記や慣用表現がどのように使われているか、実際の文献で分析するとよい。

●論理式の略記

連言 $A_1 \wedge \cdots \wedge A_n \rightarrow A_1, \dots, A_n$
 $x_1 \in S \wedge \cdots \wedge x_n \in S \rightarrow x_1, \dots, x_n \in S$

量化 $\forall x_1 \cdots \forall x_n A \rightarrow \forall x_1, \dots, x_n A$ あるいは単に A
 $\exists x_1 \cdots \exists x_n A \rightarrow \exists x_1, \dots, x_n A$

量化範囲 $\forall x (x \in S \Rightarrow A) \rightarrow \forall x \in S A$
 $\exists x (x \in S \wedge A) \rightarrow \exists x \in S A$

●量化の慣用表現

全称	$\forall x \in S P(x)$ $\forall x (Q(x) \Rightarrow P(x))$	すべての \sim は \sim である Q を満たすものはすべて P を満たす
存在	$\exists x \in S P(x)$ $\exists x (Q(x) \wedge P(x))$	ある \sim は \sim である Q を満たすもののうち P を満たすものがある
同一概念	$\forall x (P(x) \Leftrightarrow Q(x))$	P と Q は同じ性質, P と Q は同じ概念
一意	$\forall x, y \in S x = y$ $\forall x, y (P(x) \wedge P(y) \Rightarrow x = y)$	\sim を満たすものが 1 つ以下, \sim を満たすものが高々 1 つある
非一意	$\exists x, y \in S x \neq y$ $\exists x, y (P(x) \wedge P(y) \wedge x \neq y)$	\sim を満たすものが 2 つ以上ある, 異なるものが \sim を満たす
一意存在	$S \neq \emptyset \wedge \forall x, y \in S x = y$ $\exists x P(x) \wedge \forall x, y (P(x) \wedge P(y) \Rightarrow x = y)$ $\exists! x P(x)$ と略記	\sim を満たすものがただ 1 つある

論理と集合 の基本法則

(古典) 論理と集合に関する基本法則 (常に成り立つ基本性質) の一覧を示す.

なお, \top, \perp はそれぞれ 恒真, 恒偽な論理式を表す.

また, U は非空な全体集合, I は非空な添え字集合を表す.

●命題論理の基本法則

$$\begin{aligned}
 A &\Leftrightarrow A \\
 \neg\neg A &\Leftrightarrow A \\
 A \wedge A &\Leftrightarrow A \\
 A \vee A &\Leftrightarrow A \\
 A \wedge B &\Leftrightarrow B \wedge A \\
 A \vee B &\Leftrightarrow B \vee A \\
 (A \wedge B) \wedge C &\Leftrightarrow A \wedge (B \wedge C) \\
 (A \vee B) \vee C &\Leftrightarrow A \vee (B \vee C) \\
 (A \vee B) \wedge A &\Leftrightarrow A \\
 (A \wedge B) \vee A &\Leftrightarrow A \\
 (A \vee B) \wedge C &\Leftrightarrow (A \wedge C) \vee (B \wedge C) \\
 (A \wedge B) \vee C &\Leftrightarrow (A \vee C) \wedge (B \vee C) \\
 \neg(A \wedge B) &\Leftrightarrow \neg A \vee \neg B \\
 \neg(A \vee B) &\Leftrightarrow \neg A \wedge \neg B
 \end{aligned}$$

$$\begin{aligned}
 A \wedge \top &\Leftrightarrow A \\
 A \vee \perp &\Leftrightarrow A \\
 A \wedge \perp &\Leftrightarrow \perp \\
 A \vee \top &\Leftrightarrow \top \\
 A \wedge \neg A &\Leftrightarrow \perp \\
 A \vee \neg A &\Leftrightarrow \top \\
 \neg\top &\Leftrightarrow \perp \\
 \neg\perp &\Leftrightarrow \top
 \end{aligned}$$

$$\begin{aligned}
 \neg A &\Leftrightarrow A \Rightarrow \perp \\
 A \Rightarrow B &\Leftrightarrow \neg A \vee B \\
 A \Leftrightarrow B &\Leftrightarrow (A \Rightarrow B) \wedge (B \Rightarrow A)
 \end{aligned}$$

$$\begin{aligned}
 A \Rightarrow B &\Leftrightarrow \neg B \Rightarrow \neg A \\
 A \Rightarrow (B \Rightarrow C) &\Leftrightarrow A \wedge B \Rightarrow C
 \end{aligned}$$

$$\begin{aligned}
 &\text{反射} \\
 &\text{対合 (二重否定)} \\
 &\text{べき等} \\
 &\text{べき等} \\
 &\text{交換} \\
 &\text{交換} \\
 &\text{結合} \\
 &\text{結合} \\
 &\text{吸収} \\
 &\text{吸収} \\
 &\text{分配} \\
 &\text{分配} \\
 &\text{ド・モルガン} \\
 &\text{ド・モルガン}
 \end{aligned}$$

$$\begin{aligned}
 &\text{消去} \\
 &\text{消去} \\
 &\text{吸収} \\
 &\text{吸収} \\
 &\text{相補 (矛盾)} \\
 &\text{相補 (排中)} \\
 &\text{反転} \\
 &\text{反転}
 \end{aligned}$$

$$\begin{aligned}
 &\text{否定} \\
 &\text{含意} \\
 &\text{同値}
 \end{aligned}$$

$$\begin{aligned}
 &\text{対偶} \\
 &\text{複合含意}
 \end{aligned}$$

●集合の基本法則

$$\begin{aligned}
 A &= A \\
 A^{c^c} &= A \\
 A \cap A &= A \\
 A \cup A &= A \\
 A \cap B &= B \cap A \\
 A \cup B &= B \cup A \\
 (A \cap B) \cap C &= A \cap (B \cap C) \\
 (A \cup B) \cup C &= A \cup (B \cup C) \\
 (A \cup B) \cap A &= A \\
 (A \cap B) \cup A &= A \\
 (A \cup B) \cap C &= (A \cap C) \cup (B \cap C) \\
 (A \cap B) \cup C &= (A \cup C) \cap (B \cup C) \\
 (A \cap B)^c &= A^c \cup B^c \\
 (A \cup B)^c &= A^c \cap B^c
 \end{aligned}$$

$$\begin{aligned}
 A \cap U &= A \\
 A \cup \emptyset &= A \\
 A \cap \emptyset &= \emptyset \\
 A \cup U &= U \\
 A \cap A^c &= \emptyset \\
 A \cup A^c &= U \\
 U^c &= \emptyset \\
 \emptyset^c &= U
 \end{aligned}$$

●述語論理の基本法則

$\forall x P(x) \Leftrightarrow \forall y P(y)$	改名
$\exists x P(x) \Leftrightarrow \exists y P(y)$	改名
$\forall x \forall y P(x, y) \Leftrightarrow \forall y \forall x P(x, y)$	交換
$\exists x \exists y P(x, y) \Leftrightarrow \exists y \exists x P(x, y)$	交換
$\forall x (P(x) \wedge Q(x)) \Leftrightarrow \forall x P(x) \wedge \forall x Q(x)$	分配
$\exists x (P(x) \vee Q(x)) \Leftrightarrow \exists x P(x) \vee \exists x Q(x)$	分配
$\neg \forall x P(x) \Leftrightarrow \exists x \neg P(x)$	ド・モルガン
$\neg \exists x P(x) \Leftrightarrow \forall x \neg P(x)$	ド・モルガン

$\forall x P \Leftrightarrow P$
$\exists x P \Leftrightarrow P$
$\exists x (P \wedge Q(x)) \Leftrightarrow P \wedge \exists x Q(x)$
$\forall x (P \vee Q(x)) \Leftrightarrow P \vee \forall x Q(x)$

$\forall x P(x) \Rightarrow \exists x P(x)$
$\forall x (P(x) \vee Q(x)) \Leftarrow \forall x P(x) \vee \forall x Q(x)$
$\exists x (P(x) \wedge Q(x)) \Rightarrow \exists x P(x) \wedge \exists x Q(x)$
$\exists x \forall y P(x, y) \Rightarrow \forall y \exists x P(x, y)$

●集合族の基本法則

$\bigcap_{i \in I} A_i = \bigcap_{j \in I} A_j$
$\bigcup_{i \in I} A_i = \bigcup_{j \in I} A_j$
$\bigcap_{i \in I} \bigcap_{j \in I} A_{i,j} = \bigcap_{j \in I} \bigcap_{i \in I} A_{i,j}$
$\bigcup_{i \in I} \bigcup_{j \in I} A_{i,j} = \bigcup_{j \in I} \bigcup_{i \in I} A_{i,j}$
$\bigcap_{i \in I} (A_i \cap B_i) = \bigcap_{i \in I} A_i \cap \bigcap_{i \in I} B_i$
$\bigcup_{i \in I} (A_i \cup B_i) = \bigcup_{i \in I} A_i \cup \bigcup_{i \in I} B_i$
$(\bigcap_{i \in I} A_i)^c = \bigcup_{i \in I} A_i^c$
$(\bigcup_{i \in I} A_i)^c = \bigcap_{i \in I} A_i^c$

$\bigcap_{i \in I} A = A$
$\bigcup_{i \in I} A = A$
$\bigcup_{i \in I} (A \cap B_i) = A \cap \bigcup_{i \in I} B_i$
$\bigcap_{i \in I} (A \cup B_i) = A \cup \bigcap_{i \in I} B_i$

●発展学習

論理と集合の法則を代数演算の法則として捉えれば、両者の関連をより深く理解できる。ブール束 (ブール代数) や完備束などの代数系 (抽象代数) を学ぶとよい。

全称と存在 の否定

例題を通して、全称命題の否定 や 存在命題の否定 の真偽を考える。

●全称と存在 の否定の対応

対象領域が有限集合 $U = \{0, 1, 2, 3, 4, 5\}$ で、基本述語 $P(x)$ が (a) $x \leq 5$, (b) $x \neq 0$, (c) $2 \mid x$, (d) $x = 1$, (e) $x < 0$, の各例について考える。

		(1) 否定 →	(2)	(3) 否定 →	(4)
$P(x)$	$\forall x P(x)$		$\neg \forall x P(x)$	$\exists x P(x)$	$\neg \exists x P(x)$
x 0 1 2 3 4 5	(学習済み)		(新)	(学習済み)	(新)
(a) ○○○○○○	○				
(b) ×○○○○○	×			○	
(c) ○×○×○×					
(d) ×○××××					
(e) ××××××				×	

		[1] 否定 →	[2]	[3] 否定 →	[4]
$\neg P(x)$	$\forall x \neg P(x)$		$\neg \forall x \neg P(x)$	$\exists x \neg P(x)$	$\neg \exists x \neg P(x)$
x 0 1 2 3 4 5	(新)		(新)	(新)	(新)
(a) ××××××					
(b) ○×××××					
(c) ×○×○×○					
(d) ○×○○○○					
(e) ○○○○○○					

(1) (2) (3) (4)

同じ意味

[1] [2] [3] [4]

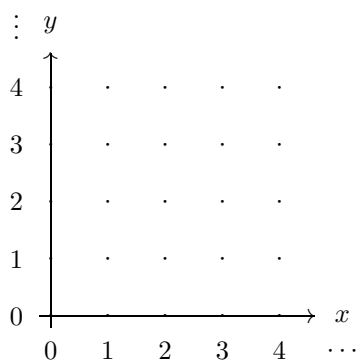
全称と存在 の併用

全称と存在を併用する命題の真偽を，例題を通して考える。
 この資料では，対象領域を非負整数全体の集合 \mathbb{N} とする。

●例 1

ある数は，どの数と比べてもそれ以下（最小の数がある）

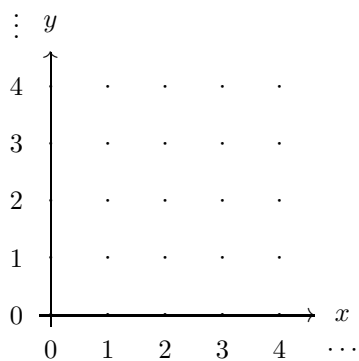
$x \leq y$ の 真 (○) 偽 (×)



●例 2

どの数にも，それ以上の偶数がある

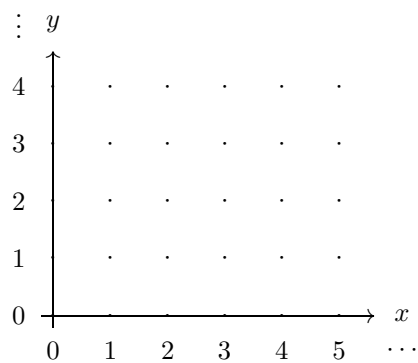
$R(x, y)$ の 真偽



●例 3 (例 2 の $\forall x$ と $\exists y$ を交換)

$$\exists y \forall x (x \leq y \wedge 2 \mid y)$$

$R(x, y)$ の真偽



式の再帰構造と木構造

式には、その内側に小さな部分式が現れるという、再帰的な (入れ子の) 構造がある。木構造を使えば、再帰構造を直観的に分かりやすく扱える。

●式の構造

「 n は正の奇数」を表す次の論理式の構文を分析する。

$$n > 0 \quad \wedge \quad \exists k \quad n = 2 \times k + 1$$

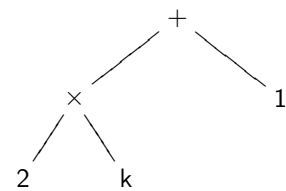
項 … 対象 (ここでは自然数) を表す式

論理式 … 真偽を表す式

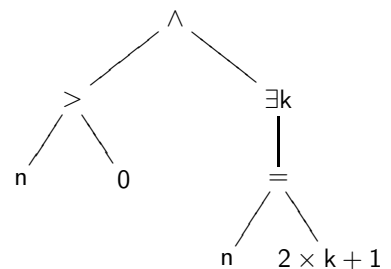
入れ子構造

木構造

項 $((2 \times k) + 1)$



論理式 $\left[\left[(n) > (0) \right] \wedge \left[\exists k \left[(n) = (2 \times k + 1) \right] \right] \right]$



式の帰納的定義

帰納的定義を、正しく並んだ記号列として式を定義するために使う。

●項の帰納的定義の例

記号 $0, 1, 2, n, k, +, \times$ から作られる項 (対象を表す式) 全体の集合は

- (1) $0, 1, 2, n, k \in T$
- (2) $t_1, t_2 \in T \implies (t_1 + t_2), (t_1 \times t_2) \in T$

を満たす記号列の (最小の) 集合 T として、帰納的に定義される。

●論理式の帰納的定義の例

上で定義した集合 T に属す項と記号 $=, >$ から作られる論理式 (真偽を表す式) 全体の集合は

- (1) $t, u \in T \implies (t = u), (t > u) \in F$
- (2) $A, B \in F \implies (\neg A), (A \wedge B), (A \vee B), (A \implies B), (A \Leftrightarrow B) \in F$
- (3) $x \in \{n, k\}, A \in F \implies (\forall x A), (\exists x A) \in F$

を満たす記号列の (最小の) 集合 F として、帰納的に定義される。

自然演繹における 仮定の有効範囲 と 変数条件

自然演繹による証明で誤りやすい, 仮定の有効範囲 と 変数条件 について理解を深める.

●仮定の有効範囲 (まとめ)

$$\frac{\begin{array}{c} i \\ A \\ \vdots \\ B \end{array}}{A \Rightarrow B} \Rightarrow I \ i \qquad \frac{\begin{array}{c} i \quad j \\ A \quad B \\ \vdots \quad \vdots \\ A \vee B \quad C \quad C \end{array}}{C} \vee E \ i, j$$

$$\frac{\begin{array}{c} i \\ \neg A \\ \vdots \\ \perp \\ A \end{array}}{A} \perp_c \ i \qquad \frac{\begin{array}{c} i \\ A[x := a] \\ \vdots \\ \exists x A \quad B \end{array}}{B} \exists E \ i$$

●変数条件 (まとめ)

$$\frac{\begin{array}{c} k' \\ k \qquad j \\ \hline \vdots \\ A[x := a] \\ \forall x A \end{array}}{\forall I} \qquad \frac{\begin{array}{c} k \qquad k' \\ j \qquad A[x := a] \qquad j' \\ \hline \vdots \\ \exists x A \quad B \end{array}}{\exists E \ i}$$

構造による式の解釈

言語は式に使う記号の集合であり、構造は記号列に意味を与えるための枠組みである。

●言語と構造の例

言語 \mathcal{L}

以下の記号からなる集合を考える。

- c は 0 変数の関数記号
- f は 1 変数の関数記号
- g は 2 変数の関数記号
- P は 1 変数の述語記号

構造 $\mathcal{A} = (U, I)$

対象領域 U が自然数全体で解釈 I が以下のものを考える。

- c^I は自然数ゼロ
- f^I は自然数上の 1 を足す関数
- g^I は自然数上の和の関数
- P^I は自然数上の「偶数である」という述語

●構造のもとでの論理式の真偽

上記の言語 \mathcal{L} に対する構造 \mathcal{A} のもとで、以下の論理式を解釈する。

$$(1) \quad P(f(f(c)))$$

$$(2) \quad \forall x P(f(x))$$

$$(3) \quad \exists x (\neg P(x) \wedge P(g(x, x)))$$