ソフトウェア基礎論 配布資料(3)

五十嵐 淳

平成 15 年 10 月 21 日

3 様々な帰納法による証明

$$\left.\begin{array}{l}\text{数学的帰納法}\\ \text{構造的帰納法}\\ \text{導出に関する帰納法}\end{array}\right\} \Longrightarrow \text{整礎的帰納法の一種}$$

3.1 数学的帰納法 (mathematical induction)

「任意の自然数に関して述語 P が成立する」の証明技法:

$\forall n \in \mathbb{N}.P(n)$ を示すには

- $P(0)$ が成立
- 任意の自然数 m に対し，$P(m)$ が成立するなら $P(m+1)$ が成立することがいえる

ことを示せば十分である．つまり,

$$(P(0)\ \&\ (\forall m \in \mathbb{N}.P(m) \Rightarrow P(m+1))) \Rightarrow \forall n \in \mathbb{N}.P(n)$$

$P(m)$ を**帰納法の仮定**(*induction hypothesis*) と呼び，$\forall m \in \mathbb{N}.P(m) \Rightarrow P(m+1)$ を induction step と呼ぶ.

この方法以外に

$$(\forall n \in \mathbb{N}.(\forall m < n.Q(m)) \Rightarrow Q(n)) \Rightarrow \forall n \in \mathbb{N}.Q(n)$$

という帰納法 (complete induction, course-of-values induction とも呼ばれる) があるが，$P(n)$ として $Q(0)\ \& \cdots \&\ Q(n)$ と考えると，前者から後者を導くことができる.

3.2 構造的帰納法

「任意の算術式 a に関して $P(a)$ が成立する」の証明技法:

$\forall a \in \mathbf{Aexp}.P(a)$ を示すには

- 任意の $m \in \mathbf{Num}$ に対して $P(m)$ が成立

- 任意の $X \in \mathbf{Loc}$ に対して $P(X)$ が成立
- 任意の算術式 a_0, a_1 に対して，$P(a_0), P(a_1)$ が成り立つという仮定の下で，$P(a_0+a_1)$ が成立
- 任意の算術式 a_0, a_1 に対して，$P(a_0), P(a_1)$ が成り立つという仮定の下で，$P(a_0-a_1)$ が成立
- 任意の算術式 a_0, a_1 に対して，$P(a_0), P(a_1)$ が成り立つという仮定の下で，$P(a_0\times a_1)$ が成立

することを示せばよい.

論理記号を使って,

$$
\begin{array}{l}
(\forall m \in \mathbf{Num}.P(m)) \,\&\, (\forall X \in \mathbf{Loc}.P(X))\& \\
(\forall a_0, a_1 \in \mathbf{Aexp}.P(a_0) \,\&\, P(a_1) \Rightarrow P(a_0 + a_1))\& \\
(\forall a_0, a_1 \in \mathbf{Aexp}.P(a_0) \,\&\, P(a_1) \Rightarrow P(a_0 - a_1))\& \\
(\forall a_0, a_1 \in \mathbf{Aexp}.P(a_0) \,\&\, P(a_1) \Rightarrow P(a_0 \times a_1)) \\
\Rightarrow \\
\forall a \in \mathbf{Aexp}.P(a)
\end{array}
$$

と書ける.

定理 3.1 (算術式の評価は決定的 (deterministic)) 任意の算術式 a, 状態 σ, 整数 m, m' に対して,

$$\langle a, \sigma\rangle \to m \,\&\, \langle a, \sigma\rangle \to m' \Rightarrow m = m'$$

証明: a の構造に関する帰納法. □

3.3 導出に関する帰納法

以下，導出規則がすでに与えられていると仮定する.「結論 A の任意の導出 $\mathcal{D}$ に関して $P(\mathcal{D})$ が成立する」の証明技法:

$\forall \mathcal{D}.P(\mathcal{D})$ を示すには

- $\mathcal{D}$ の任意の部分導出 $\mathcal{D}'$ に対して $P(\mathcal{D})$ が成立するという仮定の下で，$P(\mathcal{D})$ が成立

することを示せばよい.

この証明法を結論を明示して「A の導出に関する帰納法」という．実際には，A を導出する方法は限られているため，可能な導出の形に関する場合分けを伴なうのが典型的な証明パターンである．導出に関する帰納法は構造的帰納法を導出の構造に適用したものとしても考えられる.

定理 3.2 (コマンドの実行は決定的 (deterministic)) c をコマンド，σ を状態とする．任意の状態 σ_0, σ_1 に対し，もし $\langle c, \sigma\rangle \to \sigma_0$ かつ $\langle c, \sigma\rangle \to \sigma_1$ ならば，$\sigma_0 = \sigma_1$ である.

証明: $\langle c, \sigma\rangle \to \sigma_0$ の導出に関する帰納法．□

定理 3.1 も，評価関係の導出に関する帰納法で証明することができ，ほぼ同じ証明になる．これは，評価関係の導出がそもそも算術式の構造と対応しているためである．一方，定理 3.2 をコマンドの構造に関する帰納法で (素直に) 証明することはできない．(なぜか？)

3.4 整礎的帰納法

数学的帰納法も構造的帰納法も導出に関する帰納法も**整礎的帰納法**(*well-founded induction*) と呼ばれる，より一般的な証明技法のインスタンスである．

集合 A 上の関係 $\prec$ で，無限下降列

$$\cdots \prec a_i \prec \cdots \prec a_1 \prec a_0$$

が存在しないものを**整礎的関係**(*well-founded relation*) という．

整礎的関係 $\prec$ の反射的閉包を $\preceq$ と書く．定義は以下の通り:

$$a \preceq b \iff a = b \,\text{or}\, a \prec b$$

命題 3.1 $\prec$ を A 上の二項関係 ($\subseteq A \times A$) とする．$\prec$ が整礎的であることと，A の，任意の空でない部分集合 Q が**極小元**(*minimal element*) を持つこととは同値である．m が (順序付) 集合 Q の極小元である，とは

$$m \in Q \,\&\, \forall b \prec m.b \notin Q$$

と定義する．

整礎的帰納法の原理 $\prec$ を A 上の整礎的関係，P を A の元に関する述語とする．

$$(\forall a \in A.(\forall b \prec a.P(b)) \Rightarrow P(a)) \iff \forall a \in A.P(a)$$

例: 数学的帰納法 自然数上の (整礎的) 関係 $\prec$ を「次の数」の関係: $n \prec m \iff m = n + 1$ とすると，対応する整礎的帰納法は数学的帰納法である．

例: complete induction 自然数上の (整礎的) 関係 $\prec$ を「未満である」関係 $<$ とすると，対応する整礎的帰納法は complete induction である．

例: 算術式の構造的帰納法 **Aexp** 上の (整礎的) 関係 $\prec$ を「直接の部分式である」関係 ($a_0 \prec a \iff (\exists a_1.a = a_0 + a_1)$ or $(\exists a_1.a = a_1 + a_0)$ or $\cdots$) とすると，対応する整礎的帰納法は算術式の構造的帰納法である．

例: 導出に関する帰納法 部分導出関係 $\mathcal{D}' \prec \mathcal{D}$:

$$\mathcal{D}' \prec_1 \mathcal{D} \iff \frac{\cdots \quad \mathcal{D}' \quad \cdots}{\mathcal{D}}$$

$\prec \overset{\text{def}}{=} \prec_1^+$ (推移的閉包) とすると，対応する整礎的帰納法は導出に関する帰納法になる．

整礎的帰納法の応用　整礎的帰納法はプログラムの停止性を証明するのに，非常に有効な手段のひとつである．

定理 3.3 **IMP** プログラム Euclid を

$$\begin{aligned}\mathrm{Euclid} \equiv\ & \textbf{while } \neg(M = N) \textbf{ do} \\ & \quad \textbf{if } M \leq N \\ & \quad \textbf{then } N := N - M \\ & \quad \textbf{else } M := M - N\end{aligned}$$

とする．任意の状態 σ について，

$$\sigma(M) \geq 1 \,\&\, \sigma(N) \geq 1 \Rightarrow \exists \sigma'.\langle \mathrm{Euclid}, \sigma\rangle \to \sigma'$$

が成立する．

定理 3.4 任意の状態 σ, σ' について，

$$\neg(\langle \textbf{while true do skip}, \sigma\rangle \to \sigma')$$

3.5　帰納的な関数定義

例: 算術式の長さ

$$\mathrm{length}(a) = \begin{cases} 1 & \text{if } a \equiv n \\ 1 & \text{if } a \equiv X \\ 1 + \mathrm{length}(a_0) + \mathrm{length}(a_1) & \text{if } a \equiv a_0 + a_1 \\ 1 + \mathrm{length}(a_0) + \mathrm{length}(a_1) & \text{if } a \equiv a_0 - a_1 \\ 1 + \mathrm{length}(a_0) + \mathrm{length}(a_1) & \text{if } a \equiv a_0 \times a_1 \end{cases}$$

例: コマンドによって代入されるロケーションの集合

$$\mathrm{loc}_L(c) = \ldots$$

4　集合上の関数としての規則インスタンスと最小不動点

BNF による文法の定義，規則による関係の定義は，帰納的な集合定義の一種である．この節では，前節の内容を集合の記法で見直し，後に必要になるいくつかの概念を導入する．もっとも重要なもののひとつは**最小不動点**(*least fixed point*) の概念である．

記法 4.1 規則インスタンスを (X/y) (X は仮定の集合で y が結論) と書く．(X が空集合の場合は公理のインスタンスである．)

定義 4.1 R を規則インスタンスの集合とする．I_R を規則によって導出される結論の集合とする．つまり，

$$I_R \stackrel{\mathrm{def}}{=} \{x \mid \exists \mathcal{D}.\mathcal{D} \Vdash_R x\}$$

($\mathcal{D} \Vdash_R x$ は $\mathcal{D}$ が R を使って x を結論とする導出であることの略記である．)

定義 4.2 R を規則インスタンスの集合とする．集合 Q が R **に関して閉じている**$(R\text{-}closed)$ であるとは，任意の規則インスタンス (X/y) に対し，

$$X \subseteq Q \Rightarrow y \in Q$$

が成立することと定義する．

命題 4.1 I_R は R に関して閉じている集合の中で最小である．つまり，R を規則インスタンスの集合とすると，以下が成立する．

1. I_R は R に関して閉じている．
2. Q が R に関して閉じているならば $I_R \subseteq Q$.

この性質から導出に関する帰納法の原理が導ける．

規則インスタンスの集合は，既に得られた結論の集合から，規則を使って新しく得られる結論の集合への関数と見ることもできる．

定義 4.3 規則インスタンスの集合 R から，集合上の関数 $\widehat{R}$ を以下のように定義する．

$$\widehat{R}(B) = \{y \mid \exists X \subseteq B.(X/y) \in R\}$$

命題 4.2 集合 B が R に関して閉じていることと，$\widehat{R}(B) \subseteq B$ であることとは同値である．

$\widehat{R}$ は以下の意味で**単調**$(monotonic)$ である．

$$A \subseteq B \Rightarrow \widehat{R}(A) \subseteq \widehat{R}(B)$$

空集合から順に $\widehat{R}$ を適用して得られる集合列 A_i を考える．

$$\begin{aligned}
A_0 &= \widehat{R}^0(\emptyset) = \emptyset \\
A_1 &= \widehat{R}^1(\emptyset) = \widehat{R}(\emptyset) \\
A_2 &= \widehat{R}^2(\emptyset) = \widehat{R}(\widehat{R}(\emptyset)) \\
&\vdots \\
A_n &= \widehat{R}^n(\emptyset) \\
&\vdots
\end{aligned}$$

A_1 は公理によって得られる結論全ての集合であり，A_{n+1} は A_n から規則を一度使って得られる結論全ての集合である．$\widehat{R}$ の単調性より，

$$A_0 \subseteq A_1 \subseteq A_2 \subseteq \cdots \subseteq A_n \subseteq \cdots$$

である．$A = \bigcup_{n\in\mathbb{N}} A_n$ と定義すると，

命題 4.3

1. A は R に関して閉じている．

2. $\widehat{R}(A) = A$

3. A は R に関して閉じている集合のうち最小である．

2 番目の性質を，A が $\widehat{R}$ の**不動点**(*fixed point*) であるという．1, 3 番目の性質より，$A = I_R$ であり，不動点のうち最小のもの，最小不動点であることがいえる．I_R のことを $\mathit{fix}(\widehat{R})$ とも書き，一般に以下で定義される．

$$\mathit{fix}(\widehat{R}) \stackrel{\text{def}}{=} \bigcup_{n \in \mathbb{N}} \widehat{R}(\emptyset)$$

不動点は，($A \rightharpoonup A$ に属するような) 関数一般に対して適用できる概念である．