ソフトウェア基礎論 配布資料(5)

五十嵐 淳

平成15年11月18日

操作的/表示的意味論を使ってプログラムの性質の議論をするためには，数学的に定義された実行関係・意味関数などを対象として，メタ言語 (日本語＋集合の基礎知識) によって非形式的に議論を行なわなければならなかった．ここでは，プログラムの性質を直接の対象として扱う形式的体系 (規則による導出体系) を考える．そして，形式的体系により証明できる性質を以て，プログラムの意味を定義した，と考える，このようなプログラムへの意味付けの方法を公理的意味論と呼ぶ．これは，記号論理において各論理結合子 (「かつ」「または」など) の意味を公理 (「A & B から A を導いてもよい」など) で与えるのと似ている．公理的意味論における推論 (証明) 規則は，主要な研究者である R.W. Floyd, C.A.R. Hoare の名前をとって Floyd-Hoare 規則とも呼ばれる．

6 IMP の公理的意味論

6.1 アイデア

以下は，$1+2+\cdots+100$ を計算し，その結果を S に格納する **IMP** プログラムである．

$$S := 0; N := 1;$$
$$(\textbf{while } \neg(N = 101) \textbf{ do } S := S + N; N := N + 1)$$

では，このプログラムが本当に 1 から 100 までの和を計算していることは，どのように示せば良いだろうか．操作的意味論を用いて，終了状態で S に 5050 が格納されているという実行関係を導出することができるだろう．しかし，上のプログラムの “101” を “$P+1$” としたようなプログラムを考えたとき，そのプログラムが $1+2+\cdots+P$ を計算するという一般的な性質を示すのは P のとりうる値が無限にある限り不可能である．よって，もう少し抽象度の高い推論をすることが必要になる．

表明と不変式　そこで，プログラムの各個所で，各ロケーションの値がどのような性質を満す (と期待している) かを，コメントの形でプログラム中に埋めてみる．

$$S := 0; N := 1;$$
$$\{S = 0 \wedge N = 1\}$$
$$(\textbf{while } \neg(N = 101) \textbf{ do } S := S + N; N := N + 1)$$
$$\{S = \textstyle\sum_{1 \le m \le 100} m\}$$

$\{\ldots\}$ の部分が埋め込まれたコメントで，ロケーションの内容に関する論理式になっている．各論理式は，**while** の直前では $S = 0$ かつ $N = 1$ であること，**while** の直後 (実行終了時) には $S = \sum_{1 \le m \le 100} m$

であることをそれぞれ述べている．公理的意味論の枠組みでは，これらの論理式を**表明 (式)**(*assertion*) と呼ぶ．

さて，これらの表明に関して明らかにわかることは，**while** ループの直後では $N = 101$ が成り立っていることである．では，S についてなにか言えないだろうか？S については，少し考えてみると，各繰り返しの直後で，$S = 1 + 2 + \cdots + (N-1)$ が成り立っていることがわかる．そして，$N = 101$ と合わせて考えると $S = 1 + 2 + \cdots + 100$ が得られる．この，繰り返し (の条件判断の瞬間) において常に成立する表明を**不変式**(*invariant*) と呼ぶ．不変式は繰り返しを伴うプログラムの性質を考える上で鍵になることが多い．

部分正当性表明　公理的意味論の目標は，上のように非形式的に行っていた表明に関する推論を規則を用いた形式的推論として行うことである．上の議論をふまえて，規則の結論として導かれるものとして

$$\{A\}c\{B\}$$

という形のものを考える．ここで，A, B は表明，c はコマンドである．そして，この結論の直感的な解釈は，

> 「表明 A を満す全ての状態 σ に対して，c の実行が終了したならばその終了状態 σ' は表明 B を満す」

ということとする．この $\{A\}c\{B\}$ という形の式は**部分正当性表明**(*partial correctness assertion*) と呼ばれる．“partial” であるのは，停止しないプログラムについては何も述べていないからである．極端な場合

$$c \equiv \textbf{while true do skip}$$

である場合，$\{A\}c\{B\}$ がどんな表明 A, B に対しても成立する．

以降では，表明式の形式的な定義と意味から始め，部分正当性表明のより厳密な (表示的意味論に基づく) 意味，そして証明規則を見ていく．

6.2　表明式の言語 Assn

拡張算術式の集合 **Aexpv** を $i, j, k, \ldots$ を整数変数 (を表すメタ変数) として，以下のように定義する．

$$a \in \textbf{Aexpv} ::= n \mid X \mid i \mid a_0 + a_1 \mid a_0 - a_1 \mid a_0 \times a_1$$

表明式の集合 **Assn** を以下のように定義する．ここで，A は表明式を表すメタ変数である．

$$A ::= \textbf{true} \mid \textbf{false} \mid a_0 = a_1 \mid a_0 \leq a_1 \mid A_0 \wedge A_1 \mid A_0 \vee A_1 \mid \neg A \mid A_0 \Rightarrow A_1 \mid \forall i.A \mid \exists i.A$$

IMP の真偽値式の集合 **Bexp** は **Assn** に含まれていることに注意．

6.2.1　自由変数と束縛変数

$\forall i.A$, $\exists i.A$ において，A を i の**有効範囲**(*scope*) という．有効範囲内に出現する i を，**束縛されている**(*bound*)，といったり，i は**束縛変数**(*bound variable*) であるといったりする．また，逆に，束縛

されていない変数の出現を，**自由である**(*free*)，といったり，その変数は**自由変数**(*free variable*) である，といったりする．$FV(a)$ を a 内の自由変数の集合とすると，FV は以下のように定義できる．

$$FV(n) = FV(X) = \emptyset \qquad FV(i) = \{i\}$$
$$FV(a_0 + a_1) = FV(a_0 - a_1) = FV(a_0 \times a_1) = FV(a_0) \cup FV(a_1)$$

同様に，表明式 A に現れる自由変数の集合 $FV(A)$ は，

$$FV(\mathbf{true}) = FV(\mathbf{false}) = \emptyset \qquad FV(a_0 = a_1) = FV(a_0 \leq a_1) = FV(a_0) \cup FV(a_1)$$
$$FV(A_0 \wedge A_1) = FV(A_0 \vee A_1) = FV(A_0 \Rightarrow A_1) = FV(A_0) \cup FV(A_1) \qquad FV(\neg A) = FV(A)$$
$$FV(\forall i.A) = FV(\exists i.A) = FV(A) \setminus \{i\}$$

と定義できる．A が自由変数を持たない，つまり $FV(A) = \emptyset$ であるとき，A は**閉じている**(*closed*) という．

6.2.2 代入

自由変数を持つ表明式は「性質」を表している述語と考えられる．この自由変数に具体的な (自由変数のない) 式を当てはめることで，その式が「性質」を満すかどうかが判断できるようになる．この，表明式内の自由変数を別の算術式で置き換える操作を，**代入**(*substitution*) という．

一般的な代入の定義では，代入される式に自由変数が含まれる場合に，それが束縛されないように気をつける必要がある．例えば，

$$\exists i.i \times j = 5$$

という表明式は 「j は 5 の約数である」ことを示している．そこで「$i+1$ は 5 の約数である」という表明式を得るために．j に $i+1$ という算術式を代入することを考える．しかし，この代入を素朴に行うと，

$$\exists i.i \times (i+1) = 5$$

という期待した意味ではない式になってしまう．この問題は，通常，束縛変数の名前替えというテクニックを使って避けている．(代入結果として $\exists k.k \times (i+1) = 5$ という表明式が得られる．)

ここでは，代入操作を，**自由変数を持たない**算術式の代入に限って考える．A 中の自由な整数変数 i への a の代入 $A[a/i]$ は以下のように定義する．

$$n[a/i] \equiv n \qquad X[a/i] \equiv X \qquad i[a/i] \equiv a$$
$$j[a/i] \equiv j \quad \text{if } i \neq j \qquad (a_0 + a_1)[a/i] \equiv (a_0[a/i] + a_1[a/i])$$
$$(a_0 - a_1)[a/i] \equiv (a_0[a/i] - a_1[a/i]) \qquad (a_0 \times a_1)[a/i] \equiv (a_0[a/i] \times a_1[a/i])$$
$$\mathbf{true}[a/i] \equiv \mathbf{true} \qquad \mathbf{false}[a/i] \equiv \mathbf{false}$$
$$(A_0 \wedge A_1)[a/i] \equiv (A_0[a/i] \wedge A_1[a/i]) \qquad (A_0 \vee A_1)[a/i] \equiv (A_0[a/i] \vee A_1[a/i])$$
$$(A_0 \Rightarrow A_1)[a/i] \equiv (A_0[a/i] \Rightarrow A_1[a/i]) \qquad (\neg A)[a/i] \equiv \neg(A[a/i])$$
$$(\forall i.A)[a/i] \equiv \forall i.A \qquad (\forall j.A)[a/i] \equiv \forall j.(A[a/i]) \quad \text{if } i \neq j$$
$$(\exists i.A)[a/i] \equiv \exists i.A \qquad (\exists j.A)[a/i] \equiv \exists j.(A[a/i]) \quad \text{if } i \neq j$$

同様に，表明式 A 中のロケーション X に自由変数を持たない算術式 a を代入する操作 $A[a/X]$ も定義できる．

6.3 表明式の意味論

既に見たように，表明式は状態に関しての性質を述べたものであるので，表明式の意味は，その表明式 A を「満す」ような状態の集合として与えられる．もしくは，状態と表明式の関係と見ることも可能である．表明式に自由な整数変数を含む場合は，整数変数に具体的な整数を割り当てたもとで，状態が表明式を満すかどうかを判断する．この，整数変数への整数の割り当てを**解釈**(*interpretation*)と呼ぶ．以後，I は解釈を表すメタ変数で，$\mathbf{Intvar} \to \mathbf{Num}$ の要素である．

算術式の意味関数　表明式の意味を定義する前に，算術式の意味関数を，解釈を用いて自由変数を扱えるように拡張する．算術式 a に対し，$\mathcal{A}v[\![a]\!]I\sigma$ を解釈 I，状態 σ の下での a の表示とし，以下のように定義する．

$$\begin{array}{lll}
\mathcal{A}v[\![n]\!]I\sigma = n & \mathcal{A}v[\![X]\!]I\sigma = \sigma(X) & \mathcal{A}v[\![i]\!]I\sigma = I(i) \\
\mathcal{A}v[\![a_0 + a_1]\!]I\sigma = \mathcal{A}v[\![a_0]\!]I\sigma + \mathcal{A}v[\![a_1]\!]I\sigma & & \\
\mathcal{A}v[\![a_0 - a_1]\!]I\sigma = \mathcal{A}v[\![a_0]\!]I\sigma - \mathcal{A}v[\![a_1]\!]I\sigma & \mathcal{A}v[\![a_0 \times a_1]\!]I\sigma = \mathcal{A}v[\![a_0]\!]I\sigma \times \mathcal{A}v[\![a_1]\!]I\sigma &
\end{array}$$

表明式の意味　上で述べたように，表明式の意味は，状態，解釈，算術式の三項関係 $\sigma \models^I A$ として記述し，状態 σ は，解釈 I のもとで A を満す，もしくは状態 σ，解釈 I のもとで A が真であることを意味する．解釈 I は，A 中の自由変数の値を決定するのに使われる．また，状態の集合 Σ として「終了しない計算の (仮想的な) 到達状態」を表現する要素 $\perp$ を加えた $\Sigma_\perp \stackrel{\text{def}}{=} \Sigma \cup \{\perp\}$ を使用する．

$\sigma \models^I A$ は A の構造に関する帰納法を使って，以下のように定義する．以下で，$I[n/i]$ は，I 中の i への割り当てのみを n に変えたような環境を表す．(定義は $\sigma[n/X]$ などと同様)

$$\begin{array}{ll}
\perp \models^I A & \\
\sigma \models^I \mathbf{true} & \\
\sigma \models^I (a_0 = a_1) & \text{if } \mathcal{A}v[\![a_0]\!]I\sigma = \mathcal{A}v[\![a_1]\!]I\sigma \\
\sigma \models^I (a_0 \le a_1) & \text{if } \mathcal{A}v[\![a_0]\!]I\sigma \le \mathcal{A}v[\![a_1]\!]I\sigma \\
\sigma \models^I (A_0 \wedge A_1) & \text{if } \sigma \models^I A_0 \text{ かつ } \sigma \models^I A_1 \\
\sigma \models^I (A_0 \vee A_1) & \text{if } \sigma \models^I A_0 \text{ または } \sigma \models^I A_1 \\
\sigma \models^I \neg A & \text{if } \sigma \models^I A \text{ ではない} \\
\sigma \models^I A \Rightarrow B & \text{if } \sigma \models^I A \text{ ではない，または } \sigma \models^I B \\
\sigma \models^I \forall i.A & \text{if すべての } n \in \mathbf{Num} \text{ に対して} \sigma \models^{I[n/i]} A \\
\sigma \models^I \exists i.A & \text{if ある } n \in \mathbf{Num} \text{ に対して} \sigma \models^{I[n/i]} A
\end{array}$$

以下の命題の示すように，真偽値式 (表明式の部分集合でもある) の意味関数と，上の表明式の意味は対応している．

命題 6.1 $b \in \mathbf{Bexp}$, $\sigma \in \Sigma$ とする．任意の解釈 I に対して，

$$(\mathcal{B}[\![b]\!]\sigma = \mathbf{true} \iff \sigma \models^I b) \,\&\, (\mathcal{B}[\![b]\!]\sigma = \mathbf{false} \iff \sigma \not\models^I b)$$

表明式がある解釈のもとで真となる状態の集合を A^I と表記し，以下のように定義する．

$$A^I \stackrel{\text{def}}{=} \{\sigma \in \Sigma \mid \sigma \models^I A\}$$

部分正当性表明　さて，以上の定義を使って $\{A\}c\{B\}$ の意味は，$\sigma \models^I \{A\}c\{B\}$ という関係で示され，

$$\sigma \models^I \{A\}c\{B\} \iff (\sigma \models^I A \Rightarrow \mathcal{C}[\![c]\!]\sigma \models^I B)$$

と定義する．ただし，ここでの意味関数 $\mathcal{C}[\![c]\!]$ は $\Sigma \to \Sigma^\perp$ とする．

妥当性　我々が主に興味があるのは，特定の状態や解釈における部分正当性表明の真偽ではなく，任意の状態・解釈での真偽である．部分正当性表明が**妥当**$(valid)$ であることを $\models \{A\}c\{B\}$ と表記し，

$$\models \{A\}c\{B\} \iff \forall \sigma \in \Sigma, I \in \mathbf{Intvar} \to \mathbf{Num}.\sigma \models^I \{A\}c\{B\}$$

と定義する．同様に，表明式 A が，任意の状態・解釈の下で真であることを $\models A$ と表記する．

6.4　部分正当性表明のための証明規則

以下は，部分正当性表明のための導出規則 (Floyd-Hoare 規則とも呼ばれる) である．

$$\frac{}{\{A\}\mathbf{skip}\{A\}} \qquad \text{(H-Skip)}$$

$$\frac{}{\{B[a/X]\}X := a\{B\}} \qquad \text{(H-Assign)}$$

$$\frac{\{A\}c_0\{C\} \qquad \{C\}c_1\{B\}}{\{A\}c_0; c_1\{B\}} \qquad \text{(H-Seq)}$$

$$\frac{\{A \wedge b\}c_0\{B\} \qquad \{A \wedge \neg b\}c_1\{B\}}{\{A\}\mathbf{if}\ b\ \mathbf{then}\ c_0\ \mathbf{else}\ c_1\{B\}} \qquad \text{(H-If)}$$

$$\frac{\{A \wedge b\}c\{A\}}{\{A\}\mathbf{while}\ b\ \mathbf{do}\ c\{A \wedge \neg b\}} \qquad \text{(H-While)}$$

$$\frac{\{A'\}c\{B'\} \qquad (\models (A \Rightarrow A') \text{ かつ } \models (B' \Rightarrow B))}{\{A\}c\{B\}} \qquad \text{(H-Conseq)}$$

$\{A\}c\{B\}$ が上の規則を用いて導出可能であるとき，$\vdash \{A\}c\{B\}$ と表記する．

6.5　Floyd-Hoare 規則を使った証明例

$$w \equiv \mathbf{while}\ (N < 101)\ \mathbf{do}\ (S := S + N; N := N + 1)$$

として、

$$\vdash \{S = 0 \wedge N = 1\}w\{S = \sum_{1 \le k \le 100} k\}$$

を示す．不変式として

$$I \equiv S = N \leq 101 \wedge \sum_{1 \leq k \leq N-1} k$$

を選ぶことにすると、以下のような導出が得られる。

$$\dfrac{\dfrac{}{\{I[N+1/N][S+N/S]\}S := S + N\{I[N+1/N]\}}\ \text{H-Assign} \quad \dfrac{}{\{I[N+1/N]\}N := N + 1\{I\}}\ \text{H-Assign}}{\{S + N = \sum_{1 \leq k \leq (N+1)-1} k \wedge N + 1 \leq 101\}S := S + N; N := N + 1\{I\}}\ \text{H-Seq}$$

さらに、$\models (I \wedge N < 101) \Rightarrow (S + N = \sum_{1 \leq k \leq (N+1)-1} k \wedge N + 1 \leq 101)$ より、H-Conseq, H-While を使って

$$\dfrac{\dfrac{\vdots}{\{I \wedge N < 101\}S := S + N; N := N + 1\{I\}}\ \text{H-Conseq}}{\{I\}w\{I \wedge \neg(N < 101)\}}\ \text{H-While}$$

また、明らかに

$$\models (S = 0 \wedge N = 1) \Rightarrow I$$

と

$$\models (I \wedge \neg(N < 101)) \Rightarrow (N = 101 \wedge S = \sum_{1 \leq k \leq N-1} k)$$

つまり

$$\models (I \wedge \neg(N < 101)) \Rightarrow S = \sum_{1 \leq k \leq 100} k$$

より、H-Conseq を使って,

$$\vdash \{S = 0 \wedge N = 1\}w\{S = \sum_{1 \leq k \leq 100} k\}$$

である。

6.6 Floyd-Hoare 規則の健全性

一般に，証明システムに対しては，以下のような性質が考えられる．

- **健全性**(*soundness*): 各規則が妥当性を保存する，という性質．これにより，Floyd-Hoare 規則により導出された結論は妥当な部分正当性表明であることがいえる．
- **完全性**(*completeness*): 全ての妥当な論理式が証明システムで導出可能である，という性質．妥当な部分正当性表明の Floyd-Hoare 規則を使った証明が存在することを意味する．

ここでは，まず健全性について確認する．完全性に関しては改めて扱っていく．

補題 6.1 $a, a_0 \in \mathbf{Aexpv}$, $X \in \mathbf{Loc}$ とする．任意の解釈 I, 状態 σ に対し,

$$\mathcal{A}v[\![a_0[a/X]]\!]I\sigma = \mathcal{A}v[\![a_0]\!]I(\sigma[\mathcal{A}v[\![a]\!]I\sigma/X])$$

補題 6.2 $B \in \mathbf{Assn}$, $X \in \mathbf{Loc}$, $a \in \mathbf{Aexp}$ とする．任意の状態 $\sigma \in \Sigma$ に対し,

$$\sigma \models^I B[a/X] \iff \sigma[\mathcal{A}[\![a]\!]\sigma/X] \models^I B$$

定理 6.3 (健全性) もし $\vdash \{A\}c\{B\}$ ならば，$\models \{A\}c\{B\}$ である．