

# プログラミング言語の意味論

勝股 審也

## 1 意味論について

プログラミング言語の意味論とは、広い意味では文字列であるプログラムに何らかの「意味」を対応させる事である。普段、私たちはプログラムに対しコンパイラやインタプリタによって機械語や計算機の振舞いを対応づけ、それらを実際の計算機で動かすことで実用的な恩恵を受けている。

一方、プログラミング言語の理論的な研究においては、プログラムに対してその挙動を表現するような数学的な対象や構造を対応させ、それらを数学的な道具や知識を援用して分析することで、プログラムや言語の性質を調べるということをする。この対応づけを研究する分野を(狭い意味での)プログラミング言語の意味論と言い、これが私の講義のトピックである。

本講義ではプログラミング言語 PCF (Programming language for Computable Functions) の意味論を論じる。PCF は自然数とその上の高階関数の計算に特化したプログラミング言語である。意味論には様々なスタイルがあるが、今回は二通りの意味論を紹介する。一つは操作的意味論と呼ばれ、計算機の上でインタプリタを実装するのに似た自然な意味論である。もう一つは表示の意味論と呼ばれ、プログラムに対しその入出力関係を表現する関数を対応させる意味論である。この二つの意味論の間には「観測可能な範囲においてプログラムに同じ意味を与える」という、*adequacy* と呼ばれる関係が成り立つ。本公開講座はこの *adequacy* を目指して講義を進めたい。

## 2 表記法

自然数の集合を  $\mathbf{N}$  で表わす。集合  $X$  から  $Y$  への写像全体の集合を  $X \rightarrow Y$  で表わす。括弧が多くなると判断した場合、 $f(x)$  の括弧を省略して  $fx$  と書くことにする。この省略が続いた場合は左から順に省略が行なわれているものとする。例えば  $fx y z$  は  $((f x) y) z = ((f(x))(y))(z)$  となる。集合  $X$  の元に対して集合  $Y$  の元を一意に割り当てる対応  $x \mapsto e$  があった時、この対応が定める  $X$  から  $Y$  への写像を  $\lambda x. e$  と書く。例えば対応  $x \mapsto x^2 + x + 1$  が定める写像を  $\lambda x. x^2 + x + 1$  と書く。

## 3 プログラミング言語 PCF

### 3.1 PCF とは

この節では PCF を非形式的に説明する。PCF は型付きラムダ計算を基盤とした単純な文法を持つプログラミング言語である。本講義で紹介するのは値呼び (*call-by-value*) の PCF である。これとは別に名前呼び (*call-by-name*) の PCF というものもある。これらの区別は後程説明する。以降、 $M, N, L$  は PCF のプログラムを表わすとする。

PCF は自然数とその上の高階関数の計算に特化しており、自然数の加減と(再帰)関数の生成や呼び出しといった機構を備えている。一方で現実のプログラミング言語が備える実用的な機能、例えばファイル入出力、画面表示、ネットワークといった機器の操作や、オブジェクトやポインタ、配列といった機構は備えていない。

PCF のプログラムの実行が停止したならば結果が必ず一つ返ってくる。PCF には以下に述べるプログラム形式がある。

1. 最も単純な PCF のプログラム形式は自然数

$$\bar{0}, \bar{1}, \bar{2}, \dots$$

である(数学的な自然数と PCF の自然数を区別するため上線が引かれている)。自然数を実行するとそのものが結果として返ってくる。また、PCF には

$$\text{succ } M, \quad \text{pred } M$$

というプログラムの形式がある。これらを実行すると  $M$  の実行結果に 1 を足した / 引いた結果が返ってくる。なお  $\text{pred } \bar{0}$  の実行結果は  $\bar{0}$  である。例えば  $\text{succ}(\text{succ } \bar{5})$  というプログラムを実行すると  $\bar{7}$  が返ってくる。

実用的なプログラミング言語は自然数上の演算を豊富に持つが、PCF は  $\text{succ}$  と  $\text{pred}$  の 2 つしか持たない。しかしながら(効率の問題を除けば)これらの演算と後に述べる再帰関数形式を組み合わせることで加減乗除、もっと強く言って自然数上の計算可能な部分関数全てを表現できる。

2. 次に単純なプログラム形式は変数

$$x, y, z, \dots$$

である。プログラムの実行時、各変数には何らかの値が割り当てられており、変数を実行するとそれに割り当てられた値が結果として返ってくる。この割り当てはプログラムの実行の進展によって動的に変わる。

例えば、 $x$  に  $\bar{5}$  が割り当てられている時に  $\text{pred } x$  を実行すると  $\bar{4}$  が結果として返ってくる。

3. PCF はあるプログラムの実行結果が  $\bar{0}$  かどうかで条件分岐をする形式

$$\text{ifzero } L \text{ then } M \text{ else } N$$

を備えている。この形式を実行するとプログラム  $L$  が実行され、その結果が  $\bar{0}$  かそうでないかにより  $M$  か  $N$  の実行結果が返ってくる。例えば  $\text{ifzero}(\text{pred } x) \text{ then } M \text{ else } N$  を実行すると、 $x$  に  $\bar{0}$  または  $\bar{1}$  が割り当たっている場合は  $M$  の実行結果が、そうでない場合は  $N$  の実行結果が返ってくる。

4. PCF ではプログラムも計算結果となり得る。以下の関数形式

$$\text{fun}(x : \tau) . M$$

を実行すると、「ある値を渡されたら、それを  $x$  に割り当てて  $M$  を実行し、その結果を返すプログラム」そのものが結果として即座に返ってくる<sup>1</sup>。このプログラムは「 $x \mapsto M$  という

---

<sup>1</sup>この時  $M$  は実行されない。































