

Joseph E. Stoy : Denotational Semantic: The Scott-Strachey Approach to Programming Language Theory

MIT Press (1977)

プログラミング言語の仕様は、その言語の“構文” —つまり、どのような記号列が正当なプログラムとして受け入れられるのか —とともに、その言語の“意味”を定める必要がある。プログラミング言語にはさまざまな構成要素があり、これらの“意味”は、日本語・英語などの自然言語によって記述できる。しかし自然言語による記述は、しばしば不完全であったり矛盾を含み、実装時に誤った解釈をしやすきことは経験が示している。したがってプログラムの正当性に関する推論の根拠としては不十分であり、コンパイラの自動生成などに用いることができない。

この名著は、プログラミング言語の形式的意味を与える枠組みの1つである。表示の意味論への最初の入門書であり、Dana Scottの領域理論とChristopher Stracheyの言語記述に関する研究成果が系統的に整理され、読みやすく、明晰かつ示唆に富んでいる。著者のJoseph Stoyは、Stracheyと同じオックスフォード大学コンピューティング研究所のプログラミング研究グループに属していた。1973年にMITを訪問中、「プログラミング言語の形式的意味論」の講義ノートとして本書の初稿を書きあげた。本書は、ラムダ計算をはじめ、状態、コマンド、ジャンプ、代入を含むさまざまなプログラミング言語を扱う。当時、これらの内容が1冊の本となったのは初めてであり、プログラミング言語に関する基礎的な考え方を修得する学生や、プログラミング言語を設計する研究者にとって、最高の参考書の1つである。

表示の意味論は、プログラミング言語の意味を、集合や関数のような数学的对象で構成される数学的モデルを用いて形式的に記述しており、操作的意味論や公理の意味論とは著しく異なる。表示の意味論の特徴は、完結したプログラムに対してだけでなく、それを構成するすべての語句に対してそれぞれの意味を与える点である。プログラムの意味はそれを構成する各語句の意味だけに依存して解釈されなければならない。この原則を意味の構成原理という。実際、表示の意味論において意味関数を帰納的に定義する常套手段は、各抽象構文の解釈を表す

意味方程式の集合を与える方法である。

表示の意味論は、1964年に行われた形式言語記述言語に関する学会でStracheyが発表した論文に始まる。この論文は、抽象構文を関数へ写像する、関数の合成で定義された意味関数を導入し、不動点演算子 Y を使ってループの意味を定義している。Stracheyの初期の研究における主要な理論的問題は、表示の意味は(型なし)ラムダ計算を使って形式的に定義されるが、このラムダ計算のモデルの存在が明らかでないという点であった。

ScottはStracheyの書いてきた意味の記述法の基礎となる意味領域(モデル)の理論を構築した。意味領域は、「定義されていること(definedness)」の度合いを表す順序 \sqsubseteq が与えられた集合である。プログラミング言語の各要素を解釈する関数は、順序 \sqsubseteq を保持する意味で単調であり、そして、増加列の極限を保存する意味で連続と仮定される。連続性により、最小不動点と反射的領域を定義することができるようになる。反射的領域上の連続関数の最小不動点と解釈することで、ループや再帰的プログラムの構造の表示の意味が与えられる。また、ALGOL60における手続きパラメータを持つ手続き、LISPにおける動的束縛を持つ関数、あるいはCにおける変数への手続きの代入などのように、自己作用を含むプログラムの構造の意味を表示するには反射的領域が必要である。

プログラミング言語の意味論の多くの重要な概念を扱っている本書では、記憶(store)の形式的定義、記憶と環境との分離、反射の意味領域、そしてジャンプや例外処理を扱う継続(continuation)などが示されており、これまでのプログラミング言語の設計に大きな影響を与えてきた。

本書で扱っている内容は、その後出版された多くの優れた入門書、解説書などでカバーされているので、本書を直接読む機会はあまり多くないであろう。しかし、時々書棚から本書を取り出して埃をはらい、表示の意味論の原点に触れることは意味のあることであろう。

(平成16年9月18日受付)

胡 振江 / 東京大学大学院情報理工学系研究科
hu@mist.i.u-tokyo.ac.jp