

## Robin Milner : A Proposal for Standard ML

Conference Record of the 1984 ACM Symposium on Lisp and Functional Programming, pp.184-197 (1984).

MLは、多相型システム、パターンマッチ、モジュールシステム、参照型、例外処理などの機構を持つ関数型言語である。実用的なプログラムを記述するための汎用プログラミング言語であると同時に、プログラミング言語への新しい機能の導入、理論的考察の実証の基礎などとしても、広く使われている。

MLは、1970年代後半の定理証明システム Edinburgh LCFのメタ言語を起源とする。その後、LCFメタ言語を拡張し、種々の研究成果を採り入れて、汎用の関数型プログラミング言語として設計したものが、本論文で提案している Standard ML である。MLの歴史、特徴、基礎理論については、参考文献1)が詳しい。

1984年のLisp and Functional Programming シンポジウムの会議論文集に掲載されている本論文は、次の2点で目を引く。

まず第1に、論文集の序文によると、本論文の発表は招待されたものであったが、Milnerは都合により会議に出席できず発表もできなかったという。しかし、会議のプログラム委員会は、本論文の興味深さと、MLに関連する他発表の背景を提供しているとのことで、特例として Invited Paper として論文集に掲載することに決定したらしい。第2に、論文の1ページ目に「本稿はすでに working report としてあったが、改訂する時間がなかったため、関連研究に関するきちんとした議論がない」と弁明している点である。たしかに、本論文は仕様書あるいは提案書的な趣きがある。いずれの点も、現在の国際会議においては考え難いことであるが、関数型言語への機能導入の研究成果を採り入れ、MLを汎用的なプログラミング言語へ脱皮させようという動向に注目が集められていた時代背景を読みとることができる。

前述したように、本論文は全体的に仕様書的な印象があるが、1.2節「Design principles」には、Standard MLの興味深い設計思想が語られている。曰く、「ここで提案

する ML は、the functional language を目指してはいない」つまり、「関数型言語は ML で決まり」というものを作ろうというのではない。このことは、関数型言語にはいろいろな設計のアイデアがあり得ることを認識しており、Standard ML をその実現の1つとして捉えようという姿勢が現われている。実際その後、本会誌でプログラミングが連載されている Haskell が、ML とは異なり、遅延評価に基づく純関数型言語として設計・実装され、現在の関数型言語は大きく ML 系と Haskell 系に二分されている。Milner はそのような未来を予想していたのかもしれない。

さらに、本論文では、言語設計に関して次の3点を述べている。

1. 言語のコア部分を、simple で well-understood で well-tried なアイデアに制限する。
2. 一般化することが明らかに自然であるような well-tried なアイデアを一般化する。
3. 異なる（正しい）処理系間でのプログラム移植を最小の労力で可能となるように、言語の仕様を完全に与える。

いずれをとっても、現在でも通用するような思想が述べられており、言語設計上見習うべき部分が多い。またこのことが ML が今でも関数型の世界で中心的な言語として使われ続けている大きな要因ではなかるうか。

ML 言語の定義は、本論文の後も、さらに各種提案に関する検討がなされ、6年後の1990年に The Definition of Standard ML<sup>2)</sup> という本として出版された。

### 参考文献

- 1) 大堀 淳：ML—多相型システムを持つ関数型言語—, 情報処理, Vol.35, No.3, pp.215-226(1994).
- 2) Milner, R., Tofte, M. and Harper, R.: The Definition of Standard ML, The MIT Press(1990).

(平成17年10月24日受付)

岩崎英哉 / 電気通信大学  
iwasaki@cs.uec.ac.jp