

プログラミング言語 SML

大堀 淳

東北大学 電気通信研究所

1 はじめに

SML # は我々が開発を進めている ML 系の新しい関数型プログラミング言語である。レコード多相性、ランク 1 多相性、データベースの多相型理論、相互運用性の型理論、型主導コンパイル理論などの我々の基礎研究成果に基づく種々の機能を含む最先端の ML 系言語であるばかりでなく、C 言語や JAVA、データベースとのシームレスな連携、多バイト文字処理等のライブラリ、自動プログラミングツールなどをサポートする実用性の高い言語を目指している。本論文では、SML # の開発の背景と意図を論じた後、SML # の特徴やその理論的基礎を概観する。

2 次世代 ML 系高信頼言語の必要性

ネットワークで有機的に結合され協調して動作する大規模で高機能なソフトウェアを短時間で効率よくかつ信頼性を持って構築するためには、外部資源を動的に利用し協調して動作する複雑で高機能なソフトウェアを記述可能な、高水準高信頼プログラミング言語の活用が重要である。しかし現状は、このような要求に応え得る高信頼プログラミング言語をソフトウェア生産の現場で十分に活用できる環境になく、その結果、web サーバなどのインターネット上の基幹システムの多くは、プログラムの静的な型検査等が十分ではない言語でプログラムされ、その結果、完成した後もシステムに多くのプログラミングエラーが潜在する脆弱性をもつ。プログラムに潜在するエラーをコンパイル時に自動的に取り除く機能を持ちかつ、強力かつ柔軟な機能を備えた高信頼言語の活用は、今後予想されるインターネット等のソフトウェア開発の生産性の向上とテストや保守コストの低減に大きく寄与し、IT 産業、さらには社会の生産性の向上に大きく寄与すると考えられる。

プログラミング言語 ML は、型理論を基礎とするプログラムの静的解析技術により、プログラムの整合性

の解析と検証を行い、多くのエラーの検出をコンパイル時に自動的に行うことを実現している高信頼言語である。しかしながら、Standard ML of New Jersey や Objective Caml (OCaml) 等の現在の ML 系言語は、その基礎とする型理論の制限や処理系の実装上の制限から、実用アプリケーションの開発用言語として広く普及しているとは言いがたい。実用プログラム開発にとって特に問題と考えられるものには以下のものがある。

- 不十分なレコードの扱い

ML 系高信頼言語の最大の特徴は、その多相型型推論機構にあるが、現在の ML 系言語では、この機能はレコード構造には対応しておらず、レコード処理に関する多相型プログラムを書くことができない。レコード構造は、データベースのタプルなどの表現に使われる実用プログラムにとって最も重要なデータ構造の一つであり、レコード操作に対する多相型推論の欠如は、ML 系言語の実用化における大きな制約である。

- 相互運用性の欠如

残念ながら、ML 系言語の通常の機能（自動メモリ管理、話型プログラム、高階関数等）を実現している現在の処理系は、C などの既存言語との相互運用性を持っているとは言いがたい。例えば、最も普及している ML 系言語と思われる OCaml では、最も基本的な整数データでさえ計算機による自然な表現ではなくまた浮動小数点データは、そのまま構造体等に格納できないなどの制約がある。

今日の複雑なインターネットソフトウェアの信頼性と生産性向上の一つの鍵は、以上のような既存の ML 言語の弱点を克服し、ソフトウェア生産に利用可能な、実用性の高い次世代高信頼言語を開発することである。

3 プログラミング言語 SML # の概要

我々は、上記を含む実用上の問題点を克服しかつ最先端の機能を実現する ML 系次世代高信頼プログラミング言語実現を目指して、SML # の開発を進めている。

The Programming Language SML #

Atsushi Ohori

Research Institute of Electrical Communication, Tohoku University, 2-1-1 Katahira, Aobaku, Sendai, 980-8577, Japan
ohori@riec.tohoku.ac.jp

SML#は、既存のML言語の特徴を継承し、かつ、分散協調環境下での複雑なソフトウェアの生産性と信頼性の向上に有効な以下を含む種々の新しい機能を持つ。

- 多相型レコード演算
- ランク1多相性
- 相互運用のための多相型システム
- レコード多相を基礎とするデータベースとの連携
- 型主導コンパイルによる自然なメモリー表現

これらはいずれも、我々独自の基礎研究に基づく最先端のものであり、その理論的な有効性は確認され、ACM TODS, ACM SIGMOD, ACM TOPLAS, ACM POPL, ACM ICFP等のフォーラムで発表され(例えば [1, 6, 7, 3, 4, 9] 等)認められているものであるが、いまだに実装がなされていない最先端技術である。SML#は、ML言語の標準仕様であるStandard MLを包摂し、かつこれら最先端の研究結果を採り入れ拡張された次世代ML言語である。

これら基礎理論を基にした設計と開発により、前節で論じたレコードの扱いの制限および相互運用性の欠如の問題を含む現状のML系言語の問題点を克服している。例えば、SML#では、レコード操作を含む関数にもその最も一般的な多相型が推論され、また、ユーザはCのライブラリ直接的にリンクし、MLプログラムから呼び出すことが可能である。

4 SML#の現状と今後の展望

SML#コンパイラのプロトタイプの開発に成功し、以下のURLで公開している。<http://www.pllab.riec.tohoku.ac.jp/smlsharp/>このプロトタイプは、レコード多相性、ランク1多相性、Cとの相互運用性を含み、Standard MLの仕様を包摂する言語コンパイラである。さらに、本プロジェクトでは、SML#を使ったプログラミングのための多バイト文字処理等のライブラリや種々の自動プログラミングツールなどを開発した。現在ネイティブコードコンパイラ、JAVA及びデータベースとの相互運用機能を開発中である。

今後、さらに開発を進め、実用高信頼言語として産業界の使用に耐えうる言語として完成させていきたいと考えている。SML#はまた、高信頼言語のための基礎理論の研究や実装技術を開発する上でプラットフォームとしても有用である。すでに、SML#コンパイラの開発の過程で、コードの証明論 [2, 5] や不動点昇格による新たな関数融合方式 [8] などを含む幾つかの新しい理論や技術が産み出されている。今後、SML#

を次世代言語コンパイラの研究開発のプラットフォームとして、より安全で高機能な言語の実現を目指したいと考えている。

謝辞 SML#は文部科学省リーディングプログラム「e-Society 基盤ソフトウェアの総合開発」の一課題「プログラム自動解析に基づく高信頼ソフトウェアシステムの構築技術」の成果物である。SML#は、算譜工房(有)との共同開発による。さらに、本e-Societyプロジェクトでは、SML#コンパイラに加え、沖電気工業(株)と共同でJavaを含む高信頼プログラミング環境の開発を行った。SML#の共同研究開発者、協力者、SML#開発チームメンバーに深謝する。

参考文献

- [1] P. Buneman and A. Ohori. Polymorphism and type inference in database programming. *ACM Trans. Database Syst.* (TODS), 21(1):30-74, 1996.
- [2] T. Higuchi and A. Ohori. A static type system for JVM access control. *ACM Trans. Program. Lang. Syst.* (TOPLAS), 29(1), 2007.
- [3] A. Ohori. A compilation method for ML-style polymorphic record calculi. In *Proc. ACM POPL Symposium*, pages 154-165, 1992.
- [4] A. Ohori. A polymorphic record calculus and its compilation. *ACM Trans. Program. Lang. Syst.* (TOPLAS) 17(6):844-895, 1995.
- [5] A. Ohori. A proof theory for machine code. *ACM Trans. Program. Lang. Syst.* (TOPLAS), 29(6), 2007. 2007.
- [6] A. Ohori, P. Buneman, and V. Breazu-Tannen. Database programming in Machiavelli - a polymorphic language with static type inference. In *Proc. the ACM SIGMOD conference*, 1989:46-57.
- [7] A. Ohori and K. Kato. Semantics for communication primitives in a polymorphic language. In *Proc. ACM POPL Symposium*, 1993:99-112.
- [8] A. Ohori and I. Sasano. Lightweight fusion by fixed point promotion. In *Proc. ACM POPL Symposium*, 2007:143-154.
- [9] A. Ohori and K. Yamatodani. An interoperable calculus for external object access. In *Proc. ACM ICFP Conference*, 2002: 60-71.