

ISO 規格 ISLISP 処理系の開発

5 E - 6

新谷 義弘* 長坂 篤* 五味 弘** 高橋 順一** 梅村 恭司*** 湯浅 太一****

*: 沖電気工業 ** : 沖テクノシステムズラボラトリ *** : 豊橋技術科学大学 **** : 京都大学

1. はじめに

IIISLISP^[1] は 1997 年に制定された, ISO 規格の LISP 言語であり, 日本案がベースドキュメントとなっている. 我々は, この ISLISP の普及を目的として, 可搬性に優れ, かつ高性能な ISLISP 処理系を開発した.

本稿では, ISLISP 言語仕様の概要と今回開発した処理系の開発目的と概要について述べる.

2. ISLISP 言語の特徴

2.1 ISO 国際規格 ISLISP

LISP はその拡張性により Lisp1.5 以来, 大学および研究機関を中心に様々な言語機能拡張や言語研究のプラットフォームとしての機能拡張が行われ, MacLisp や InterLisp を初めとする多くの方言が開発されてきた. 1980 年代の AI 実用化研究において, Lisp を産業界における実用的言語とするために必要な Lisp プログラムの Portability を実現することを目標に, これまでの多くの Lisp を統合した Common Lisp^[2] が 1984 年に開発され, オブジェクト指向機能 CLOS や例外処理機能等の機能拡張を行なった第 2 版(CLL2)が 1990 年に発表された.

Common Lisp は多くの Lisp 方言の機能を統合すると共に, Lexical Principle の採用やオブジェクトシステム CLOS の導入等新しい機能を導入した高機能な Lisp であり, 産業界における標準としての地位を確立したといえる. 一方, その言語仕様の巨大さから, 実行効率や学習・利用の面からの問題が指摘されてきた.

ISLisp は, Common Lisp の抱えるこのような問題を解決し, コンパクトで効率が高く, かつ使いやすい Lisp 言語を開発することを目的として開発された. ISO

における LISP の国際標準化活動は, 1987 年に ISO/IEC JTC1 SC22 に WG16 LISP が設置されて開始された. ISLISP のベースドキュメントとして, LISP の核部分に対して伊藤, 湯浅, 梅村等の KL (Kernel LISP) が, オブジェクト指向機能に対して Common LISP の CLOS(Common Lisp Object System) をベースとした ILOS(ISLISP Object System)が採用された^[3].

ISLISP は現在 情報処理学会 ISLISP JIS 原案作成委員会において JIS 化作業が進められている^[4].

2.2 ISLISP の特徴

ISLISP は, 当初 Common Lisp のサブセットを目標として開発され, Common Lisp の基本的な言語仕様を受け継ぐと共に, Scheme を意識したコンパクトな仕様や構文の影響を受けている. LISP としての ISLISP は次のような特徴を持っている.

- コンパクトな仕様と実行効率の高さ

Common Lisp の巨大な仕様を避けるため, オブジェクトシステム, 基本的データクラス等を初めとする基本的言語機能の簡潔化, Pretty Print 機能等の周辺機能の削除が行われた. 基本データクラスからは複素数, 有理数, ハッシュ表, ランダムステート等が削除された.

- 効率の高いオブジェクトシステム

Common Lisp のオブジェクト指向機能である CLOS は, Meta Object 機能に見られる非常に高度な機能を持っている反面, 仕様の巨大さと実行効率の問題があった. ISLISP では, 多重継承は, クラス C がクラス C1 および C2 の下位クラスであるのは, C1, C2 のそれぞれの上位クラスで共通のものが <standard-object> あるいは <object> である場合に限られる. この制限により効率的な実装が可能となる.

- 言語仕様と処理系仕様の分離

LISP システムでは, 従来言語仕様と処理系仕様とが混在している場合が多かった. ISLISP では, マクロ展開等が処理が終了した ISLISP プログラムを構成する評価形式を準備された(prepared) 評

"An Implementation of ISO Lisp Standard ISLISP system,"

Yoshihiro Shintani*, Atsushi Nagasaka*, Hiroshi Gomi**, Jun-ichi Takahashi**, Kyoji Umemura***, Taiichi Yuasa****

*Oki Electric Industry Co., Ltd.

**Oki Technosystems Laboratories, Inc.

***Toyohashi University of Technology

****Kyoto University

Windows は米国 Microsoft Corporation の登録商標です.

Pentium は米国 Intel 社の登録商標です.

価形式と呼び、準備の処理方式については規定しない。

- データ型のオブジェクトシステムへの統合

Common LISP では、データ型とクラスとが混在していたが、ISLISP では全てのデータ型はクラスとして統一された。

仕様の検討が行なわれながら、今回の ISLISP 規格に含まれなかった機能として、パッケージあるいはモジュールがある。実用的なソフトウェアの開発では、名前の衝突を解決するこれらの機能は不可欠であり、今後の課題である。

3. 処理系開発の目標と構成

今回の ISLISP 処理系の開発は、ISLISP の普及を目的としており、処理方式の開発にあたっては以下の設計目標を設定した。

- 現在使用されている多くのコンピュータシステムへの移植が容易であること
- 十分な実行速度の達成
- 短期間での開発

移植性と高速性の両立は一般に困難な課題である。筆者等はこれまでにこの2つの目標を両立させた CLiL2 処理系である Tachyon Common Lisp^[1]を開発した。Tachyon Common Lisp は RISC プロセッサを対象としており、現在最も広く使用されている Intel Pentium プロセッサに適用するには使用可能なレジスタ数の制限から大幅な方式変更が必要であり、採用しなかった。以上から我々は実行速度よりも移植性と開発期間を優先し、仮想マシン上の Byte Code 方式を採用した。

図1に本 ISLISP 処理系の構成を示す。

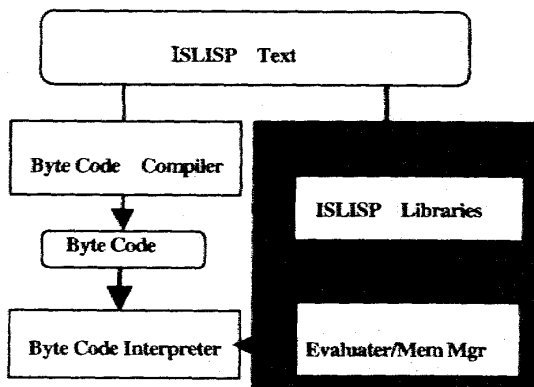


図1. 処理系の構成

以下に本 ISLISP 処理系の概要を述べる^[2]。

- Byte Code マシン

Byte Code 命令は、LISP を意識したスタックマシンとなっている。将来のネイティブコード生

成を意識したレジスタマシンも仮想マシンとして検討したが、レジスタ数の制限の強い Pentium での実行効率からスタックマシンとした。

- 8Bit 固定長タグシステム

Pointer Tag 方式による 8Bit 固定長タグ方式とした。Cons, Symbol は直接アクセス、整数 (Shortnum) は即値型、その他はヘッダ領域+データ実体の構成とし、頻繁に使用される LISP オブジェクトへのアクセスが高速となるような構成とした。

- Mark&Sweep 方式による GC

処理系は当初、現在最も広く使用されている Windows95/WindowsNT 及び UNIX(LINUX)マシンを対象として開発した。これら以外のシステムへの移植も容易である。

4. おわりに

LISP 言語の国際規格として制定された ISLISP の処理系を開発した。本処理系は ISLISP の普及を目的として、多様なコンピュータシステム上で動作するために、仮想マシンコード方式を採用した可搬性に優れたものになっている。また、実行時の効率を考慮した実装方式により、非常に高速な処理系となっている。

本処理系は、現在オブジェクト指向機能を除くインタプリタ機能と Byte Code コンパイラの基本部分が動作している。Gabriel Benchmark (一部)による性能評価では Christian Jullien による ISLISP 処理系である Open Lisp よりも高速であることが確認されている。今後、オブジェクト指向機能部分の完成を目指す計画である。

最後に、本開発は IPA 「独創的創造的ソフトウェアの開発」事業の一環として行われた。ここに関係諸氏に感謝いたします。

参考文献

- [1] ISO/IEC 13816:1997(E), Information Technology-Programming Languages, Their Environments and System Software Interfaces -Programming Language ISLISP, 1997
- [2] Guy L. Steele Jr., "Common LISP the Language, 2nd edition," Digital Press, 1990
- [3] 伊藤貴康, LISP 言語国際標準化と日本の貢献, 情報処理 38 巻 10 号, 1997 年 10 月
- [4] プログラム言語 ISLISP 作成原案, ISLISP JIS 原案作成委員会, 1997 年 12 月
- [5] Atsushi Nagasaka et al., "Tachyon Common Lisp: An Efficient and Portable Implementation of CLiL2," ACM Lisp and Functional Programming Symposium, 1992
- [6] 各務寛之他, ISO 規格 ISLISP 処理系の実装方式 情報処理学会第 56 回全国大会予稿集 5E-07, 1998 年 3 月