Ｌｉｓｐ用関数関連図の表現方法

7G-7

山守 一徳　白崎 昌俊　西垣 秀樹

(株)沖テクノシステムズ ラボラトリ

１．まえがき

Ｌｉｓｐを用いてプログラミングを行う時、生産的なプログラミングというよりもむしろ実験的なプログラミングを行なっていることが多く、システム設計、モジュール設計、プログラム設計、コーディング、プログラムテスト、システムテストといった段階をはっきり分けたWaterfall型のソフトウェア開発工程に従っていることは少ない。しかし、大規模システムを開発する時、ソフトウェアの保守・管理面から考えると、種々の仕様書などのドキュメントは、非常に大切である。我々は、これまでのドキュメントが、手続き型言語向きであることに着目し、Ｌｉｓｐなどの関数型言語に適したドキュメントについて考察した。

本稿では、関数間の呼び出し関係を図示したＬｉｓｐ用の関数関連図の表現方法について報告する。

２．関数関連図の用途

関数関連図の用途には、システム作成中のデバッグ時にプログラムの解析を支援することと、バージョンアップやプログラムの再利用、ユーザに対する保守などのためにシステム完成後のソフトウェアドキュメントとして残すことの２通りがある。

（１）ソフトウェアの保守管理

通常のソフトウェアの開発工程では、システム仕様書、モジュール仕様書、モジュール関連図をツールとして使用している。モジュール仕様書からプログラミング言語にコーディングを行なう時に、関数型言語であるＬｉｓｐを用いる場合には１つのモジュールが複数個の関数によって実現される。大規模なソフトウェア開発においては、ソフトウェアの保守・管理のために、それらの関数の呼び出し関係図が必要である。

モジュール間の関係を表わしたモジュール関連図をさらに詳細な関数レベルで表わしたのが関数関連図であるが、ソフトウェアの開発工程の中で、モジュール関連図は、コーディング前のモジュール設計時に作成されるのに対し、関数関連図は、コーディング後に作成されるものである。どちらもソフトウェアのドキュメントとして残されるものである。

（２）プログラムの解析支援

Ｌｉｓｐを用いたプログラミング環境は非常にすぐれており、デバッグにおいても会話的に容易にデバッグが行なえる。デバッガ、トレーサ、ステッパなどのツールがプログラムの動的解析を支援するものとすれば、関数関連図は、静的解析を支援するものである。InterLispにはＬｉｓｐプログラムを解析してデータベースに格納し、関数や変数の情報を表示するmaster scopeと呼ばれる関数がある。その中のshowpathコマンドでは、関数間の呼び出し関係が樹状に図示される。プログラムを解析する場合、プログラムの全体的な構造をとらえるために、関数間の呼び出し関係を調べることは必要であるが、典型的なＬｉｓｐプログラマは、Ｌｉｓｐソースプログラムを直接解読することが多い。

関数関連図はＬｉｓｐソースプログラムを解読する時の手助けをするためのものである。そのため関数間の呼び出し関係を樹状に図示するだけでなく、プログラムの基本構造の情報を付加し、ソースプログラムとできるだけ容易に対応ができるようになっている。

３．Ｌｉｓｐ用関数関連図の特徴

(1) 関数の呼び出し方法を区別する。

Ｌｉｓｐの場合には、関数の返り値を他の関数の引数に用いるために、関数呼び出しが入れ子構造となる。ユーザ定義関数が呼出されている時に、それが他のユーザ定義関数の引数として呼出されているか否かを区別する。

(2) 再帰呼び出しを図示する。

再帰呼び出しはＬｉｓｐでは頻繁に使われており、複雑なプログラム構造のもととなっている。自己再帰関数や相互再帰関数である場合は、そのことを示すマークを図示する。

(3) プログラムの基本構造を図示する。

関数定義内の繰り返し構造と条件分岐構造は、プログラミングを行なう時の思考過程にも使われる基本構造である。loop関数やcond関数などの呼び出されている場所とユーザ定義関数の呼び出されているを場所の関係を保存し、プログラムの基本構造がわかるようにする。

(4) 関数内部で定義される関数を図示する。

関数内部で定義される関数は、プログラムの静的解析では、どのように使われるのかを解析するのは難しい。その関数が定義されている関数を図示するところに付随して、内部で定義している関数を図示する。

(5) 関数の定義方法を区別する。

defun、defmacro、defmethodなど関数定義の方法は、Object指向Programmingの導入などにより種類が増えつつある。これらに対処するために、関数定義の方法を区別するマークを図示する。特に、マクロ定義関数である場合は、そのマークを図示し、ソースプログラムとの対応を容易にするために、マクロ呼び出しはマクロ展開を行わず関数の呼び出しを解析する。

(6) 出力用紙のサイズに合せて樹状出力する。

ソフトウェアドキュメントの用紙サイズまたは

A Representaion of the Calling structure for LISP functions
Kazunori Yamamori, Masatoshi Shirasaki, Hideki Nishigaki
Oki Techno Systems Laboratory, Inc.

画面サイズに合せて最大桁数と最大行数が指定でき、樹状出力が１ページ内で見やすく収まるように、適切な位置で樹状出力が中断される。

４．Ｌｉｓｐ用関数関連図の表示

関数関連図がデバッグ時に容易に扱えるためには、端末で表示ができなければならない。また、ソースプログラムと容易に対応ができるために、樹状出力は横向きに表示する。

図１に関数関連図の例を示す。

ユーザ定義の関数名は、端末上では白黒反転モードで表示される。関数名の右から出ている樹状出力は、その関数内で呼び出している関数であり、呼び出し順序に従って上から下に並んでいる。関数名の下から出ている樹状出力は、その関数の引数パラメータとして使われている関数である。プログラムのデータ構造を示すために、loop、condなどの繰り返し関数と条件分岐関数は、段下げを行なって表示している。関数名の上についているのは関数識別番号、下についているのは自己再帰関数または相互再帰関数であることを示すマーク、右についているのは既に別の場所で出力されていることを示し、その時の関数識別番号である。端末画面サイズを越えるところでは樹状出力が中断され、波括弧で囲まれたラベルによって中断箇所が示されている。

５．関数関連図作成ツール

ソースプログラムから関数関連図を作成するツールを作成し、ドキュメンテーションの省力化を行なった。このツールは、関数呼び出しを静的解析し、その結果をデータベース（属性リスト、大局変数）に格納する解析ステップと、関数を指定し、その関数を樹状出力のrootに持つ関数関連図を端末に出力する表示ステップから成る。

o関数の呼び出し情報の収集

従来のユーザ定義関数か否かの決定には、システム定義関数リストをあらかじめ作成してしておき、そのリスト中の要素でない関数をユーザ定義関数であると判断するか、そのシンボルが登録されているパッケージがユーザパッケージである時にその関数をユーザ定義関数であると判断している。

しかし、従来の決定方法ではシステム関数の解析・デバッグに用いることができない。そこで、システム関数の解析・デバッグにも用いられるように解析ステップではファイルを入力とし、そのファイルの中で定義されている関数については、ユーザ定義関数と同等の扱いを行なう。

oObject指向Programmingのサポート

最近のＬｉｓｐには、Object指向Programmingが導入され、message passing styleのプログラムが書ける。この時、関数名に相当するものは、messageのselector名であるが、class名とselector名を同時に指定しないと、一意に関数定義が定まらない。しかし、静的解析では、receiverの変数名からclass名を定めることができないので、selector名を関数名として用い、それがselector名であること示すマークを図示する。そして、再帰関数であるか否かのチェックは行わないことにする。

６．あとがき

Ｌｉｓｐプログラムのデバッグ・解析・保守・管理に役立つ関数関連図の表現方法を提案した。今日のＬｉｓｐプログラムはバージョンアップが激しく、無修正のまま長期間使われることは少ない。そのため、ソフトウェアドキュメントも的確な情報を持ち、かつ省力化されたものでなければならない。関数関連図に対する評価は今後の課題である。

最後に、日頃より熱心に御指導して頂く沖電気工業(株)総合システム研究所 椎野 努部長・長坂 篤係長はじめ、ＡＩﾜｰｸｽﾃｰｼｮﾝ第二研究室の皆様に感謝致します。

《参考文献》

(1) D.R.Barstow 他 "From Interactive to Intelligent Programming Environments." Interactive Programming Environments, Barstow et al(editors),McGrow-Hill

(2) W.Teitelman 他 Interlisp Programming environment Xerox Palo Alto Research Center,April, 1981

(3) W.Teitelman 他 The Interlisp Reference Manual,Xerox Palo Alto Research Center,1978

(4) 太田 他 "UtiLispプログラムの解析ツール", 昭和61年度電通全大,1718

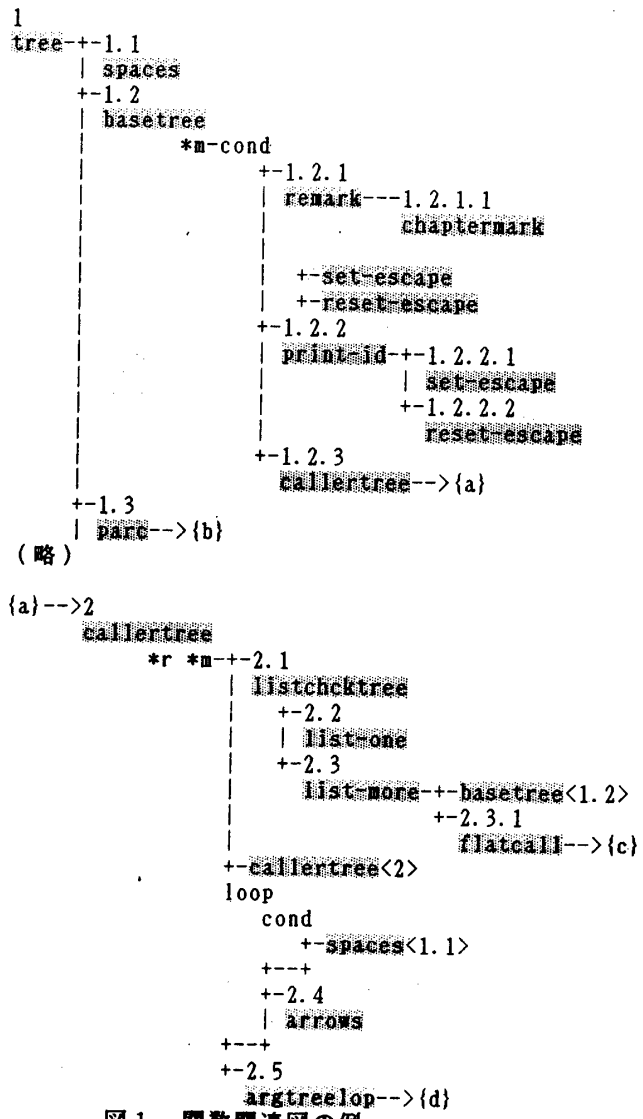

図１．関数関連図の例