2P-6

パターン記述に基づくＬＩＳＰプログラムの変換

沢井真二† ，太田義勝† ，吉田雄二† ，福村晃夫‡
（†名古屋大学） （‡中京大学）

1. はじめに

ＬＩＳＰは現在、種々の方言が存在し、必ずしも相互の互換性がよいとは言えない。本稿ではＬＩＳＰプログラムの互換性の問題に対処するため、あるＬＩＳＰの仕様に従って書かれたプログラムを他のＬＩＳＰの仕様に従うプログラムへ変換を行なう手法として、パターン記述された変換規則による変換を提案する。

2. ＬＩＳＰプログラムの変換

今回提案する手法は以下の方針により構成される。

- 与えられたＬＩＳＰの形式(form)を他のＬＩＳＰの仕様による形式に書き換える事を基本とする。
- 引数に関する仕様をパターン記述によって表現し、これに従ってＬＩＳＰプログラムを解析する。
- 引数に関する情報が記述されている部分（データ部）とプログラムの変換を実際に行なう部分（トランスレータ部）を独立させる。そして実際の変換の方法もデータ部に記述しておく。データ部を取り替えることにより、同じトランスレータが各種のＬＩＳＰ間の変換に対応出来るようにする。

3. プログラム変換の記述

プログラム変換の規則が記述されているデータ部は解析用データと合成用データよりなる。

3. 1 解析用データ

解析用データは入力されたプログラムをトランスレータが解析する際に参照するデータであり、パターン記述により表現されている。パターンの構文は図１のようになる。

```
<pattern>::=<lambda-list>
            |<repeat-pattern>
            |<optional-pattern>
            |<concatenate-pattern>
            |<list-pattern>
            |<constant-pattern>
            |<primitive-pattern>
<primitive-pattern>::=<sexpr-type>
                      |<list-type>
                      |<fun-symbol>
                      |<var-symbol>
                      |<atom-type>

<repeat-pattern>::=(repeat <pattern>)
<list-pattern>::=
          (list <pattern>…<pattern>)
<lambda-list>::=(lambda-list)
<sexpr-type>::=(sexpr-type)
<var-symbol>::=(var-symbol)
```

図１．パターンの構文

それぞれのパターンの意味は以下の通りである。

<lambda-list> … ラムダリスト（変数の設定部）

<optional-pattern> … 省略可能なパターンの並び

<concatenate-pattern> … いくつかのパターンの並び

<list-pattern> … パターンのリスト

<repeat-pattern> … パターンの繰り返し

<sexpr-type> … 一つの形式

<list-type> … コンス・セル

<atom-type> … アトム

<fun-symbol> … 関数名シンボル

<var-symbol> … 変数名シンボル

解析用データの例を図２に示す。パターン(sexpr-type)に対応する入力プログラムの部分は変換すべきＬＩＳＰの形式とみなされ、再帰的に変換が行なわれる。記述中の<…>は各パターンに対応する出力プログラムの部分を引用するための一時変数を表す。

Conversion of lisp programs based on pattern description

Shinji SAWAI†, Yoshikatsu OHTA†, Yuuji YOSHIDA†, Teruo FUKUMURA‡
† Nagoya University, ‡ Chukyo University

```
(DEFUN (fun-symbol <var1>)
       (lambda-list <var2>)
       (repeat (sexpr-type) <var3>))

(PUTPROP (sexpr-type <var1>)
         (sexpr-type <var2>)
         (sexpr-type <var3>))

(ASSQ (sexpr-type <var1>
      (sexpr-type <var2>))

(EXFILE (sexpr-type <var1>)
        (option (sexpr-type) <var2>))
```

図２．Ｓ式によるパターンの記述

３．２　合成用データ

合成用データは出力プログラムを生成する規則をＬＩＳＰの“形式”で記述したものである。図２の例に対応する合成用データを図３に示す。

```
(list 'DEFUN <var1> <var2> <var3>)

(list 'SETF
      (list 'GET <var1> <var3>)
      <var2>)

(list 'ASSOC ,<var1> ,<var2>
             ':TEST '(FUNCTION EQ))

(cons 'LOAD (cons <var1> <var2>))
```

図３．データの例

記述中の<…>は、解析部で得られた対応する入力プログラムの部分を表す。この合成用データを評価することにより変換されたプログラムが生成される。

４．変換システム

変換システムの概念図を図４に示す。

①データ解釈部　ＬＩＳＰ１の関数仕様に従って書かれたプログラムを入力データとして読み込み、解析・合成用データ部のパターン記述されたデータを参照しながら、別のＬＩＳＰ２の関数仕様に従うプログラムへと書き換え、後処理部へ送る。

②後処理部　データ解釈部の出力を受け取り、変数束縛の相違への対応、マクロ文字やエスケープ文字の処理等を行い、目的のプログラムを出力する。また、パターンの構文では変換方法を記述できないが、他の方法で変換を行なうことが可能な部分に対しての変換は後処理部で行なう。

③変換・合成用データ部　この部分は実際には各関数名シンボルの属性リストとして定義されている。

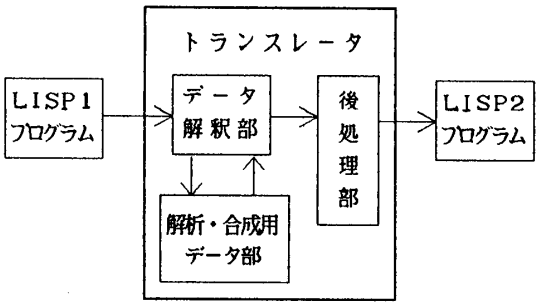

図４．変換システムの概念図

５．評価

このシステムとパターン記述を用いて、ＵＴＩＬＩＳＰにより書かれたプログラムからＣＯＭＭＯＮＬＩＳＰのプログラムへの変換を行なった。その結果を表１に示す。変換出来ない関数としては、入出力関係の関数やエラー処理関数、大型機のＯＳに依存する関数等がある。

プログラム	使用されている関数	変換された関数
"USE"	916	819
"LISPDAP"	811	794
"KALAH"	343	323

表１．変換結果

６．終わりに

本研究は名古屋大学大型計算機センターのＦＡＣＯＭ　Ｍ－７８０のＵＴＩＬＩＳＰを使用している。パターン記述の他のＬＩＳＰ間の変換への適用、後処理部の作成、及びユーザーインターフェイス等を含む環境化を今後行う予定である。

謝辞　日頃御指導頂く名古屋大学渡邉豊英助教授、ならびに吉田研究室の皆様に深く感謝します。

参考文献

(1) ＵＴＩＬＩＳＰマニュアル，富士通編

(2) Guy L. Steele Jr. :“COMMON LISP THE LANGUAGE”, Digital Press

(3) 川村他，：“ＬＩＳＰのためのパターン指向型プログラミングシステム”，信学論（Ｄ），J70-D,6,pp.1149-1156（昭62-6）