214 Lispマシン ALPS/Iの機能について

井田 昌之
(青山学院大学 理工学部)

1. イントロダクション

ALPS/Iは Lisp用パーソナルマシンである。その目標は，実際的なLispプログラムの経済的走行と会話型の専有処理形態の提供にある。ハード及びソフトの概略は文献1及び2に既に発表したが，ここではLispプロセッサのその後の改良点及び未発表の部分について報告を行なう。

2. Lispプロセッサの構成

ALPS/IのLispプロセッサはEvalquote入力を解釈実行する約2 kbyteのプログラムでPROMに書き込まれ使用される。表1に慣例に従い，標準問題の実行結果を示す。所要時間は大型機のLispに比して約5倍～50倍要するが，例えばBit A8を処理し得る事は，ALPS/Iの容量的制約が非常にゆるい事を示している。速度に関しては，第1版に比して現有プロセッサは約1.5倍の速さに改良され，マイコンCPUにもかかわらず同程度の規模のミニコンLispに匹敵するものになっている。これは次にまとめるような改良点に依存する所が大きい。

表1. 実行結果
(Sortの問題でシステム組込みのEqualをAppendを用いると時間は4分の3に短縮される)

	実行時間 (mSec)	使用セル数
Wang A	176	826
Wang B	970	1,285
Bit A6	8,750	4,738
7	29,800	15,836
8	99,500	54,514
Bit B6	1,980	1,283
7	5,950	3,476
8	19,400	11,145
Sort60	77,500	31,402
80	110,000	47,227
100	137,000	over56K

1. 変数束縛方法の変更：a-list ⇨ stack
2. アトムの構成：p-list ⇨ H分子
3. Lisp 1.6型go文を持つprog：中間言語のインタプリトとハッシングによるラベルサーチ・1.5型go文の中間言語上での無条件分岐文への置換え等を行なう高速progインタプリタ
4. 即時回収ガベジコレクタ[3]：インタプリタ内で消費するセルは通常のG.C.の作動を待たずに即時回収される。(Sortの場合，インタプリトしながら全消費セルのうち約20%を回収してしまう。) このためにevlisの定義を左図のように変更する。ここで*lastはシステム内部変数で*setq[x;y]はyを評価し関数の値とし，かつ内部変数xにそれを代入する副作用があるものとする。これによりevlis後不要になるリストmを次の操作でフリーリストに返す事ができる(現在のフリーリストのトップを保持するシステム内部変数freeltopがあるとする)。

```
evlis[m]=
 [null[m]→*setq[*last;NIL];
  null[cdr[m]]→
   *setq[*last;list[eval[car[m]]]];
  T→cons[eval[car[m]];evlis[cdr[m]]]]
```

```
rplacd[*last;freeltop]; *setq[freeltop; m];
```

5. 基本整数(small number)；0～1023の整数はアドレス自身が値となる。
6. control stack と value stack (G.C.保護用スタックと呼ぶ) の分離。

3. ALPS/Iでの付加機能

これらの処理能力向上の手段はハッシングの全面採用とバルクメモリ空間の配置の高効率化に支えられているが，その中でもH分子の役割は大きい。

1. H分子；H分子は属性部・値部・鍵部等を持ちhashingによりH-area中

にユニークに定められるもので，一般に図1の形式をしている。その構造はHLisp型H分子[3]とは全く異なり2語1組で用いられ，その属性により11種に分けられる（図2）。この事によりLisp内のユニークオブジェクトを統一的に扱うことができる。

図1　H分子

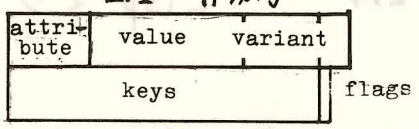

図2　H分子の分類

	atomic symbols (keys=pname)								associators (keys=(key1*.key2*))		
attribute name	undef	subr	fsubr	expr	fexpr	apval	hexpr	harray	harray element	assoc-comp associater	label associater
created by	read etc.	—	—	define deflist	deflist	cset csetq	deflist assoc-comp	array	seta	hexpr refer	prog label
deleted by	ggc	—	—	—	—	—	—	de-array	delete dearray	ggc	ggc

2．連想計算[3]；hexpr 属性が連想計算を行なう手続きに対して与えられる。このために例えばapply の定義に次の部分が付加される。（fnがアトムである所に）

```
eq[get-attrib[fn];HEXPR]→
  hdotp[fn;args]→hassoc[fn;args]; T→progn[
  v:=apply[value[fn];args];hstore[fn;args;v]]
```

3．ハッシュ化配列；harray属性のアトムをkey1に持つ連想子をハッシュ化配列要素といい，seta関数により値がセットされ，delete及びdearrayにより削除される。

```
table[i]:=v ⟹ seta[table;i;v]
v:=table[i] ⟹ setq[v;table[i]]
```

各要素の値の参照は簡単に行なえるようにevalの中のcar[form] がアトムである所に次の手続きが加えられている。

```
eq[get-attrib[car[form]];HARRAY]→
      hassoc[car[form];eval[cadr[form]]
```

例を図3に示す。ハッシュ化配列は，整数以外の任意の添字を持つ事ができ，これにより(a,v)型連想三つ組を保持する。

図3　実行例

```
FUNCTION EVALQUOTE HAS BEEN ENTERED, ---
ARRAY(GERMAN ENGLISH)

END OF EVALQUOTE, VALUE IS..
(GERMAN ENGLISH)

FUNCTION EVALQUOTE HAS BEEN ENTERED, ---
SETA(GERMAN (QUOTE WATASHI) (QUOTE ICH))

END OF EVALQUOTE, VALUE IS..
ICH

FUNCTION EVALQUOTE HAS BEEN ENTERED, ---
EVAL((GERMAN (QUOTE WATASHI)))

END OF EVALQUOTE, VALUE IS..
ICH
```

4．豊富な関数群；組み込まれたアトムを表2に示す。上述した機能のための関数（Assoccomp, Array等）に加えて，インタプリタのトレースの管理（Walk, Unwalk）・オブジェクトロード機能（*fn1, *fn2, load：RamへHexadecimal Tapeをロードし実行させる）・汎関数Try（図4）などが含められており，電源投入動作のみで快適に入力ができる。

図4 汎関数 Try

```
try[x;fn]=
 prog[[xx];
  xx:=x;
LOOP
 [null[xx]→return[xx];
  fn[car[xx]]→return[car[xx]]];
  xx:= cdr[xx];
  go[LOOP]    ]
```

表2 ALPS/I 組み込みアトム（下線部は独自のもの）

SUBR (73)	*FN1, *FN2, **GO, ADD1, APPEND, APPLY, ASSOCCOMP, ATOM, CAR, CDR, CAAR, ..., CDDDR, COMPRESS, CONS, CSET, DEFINE, DEFLIST, DIFFERENCE, DIVIDE, EQ, EQUAL, ERROR, EVAL, EVLIS, EXPLODE, GENSYM, GREATERP, LESSP, LOAD, MAP, MAPCAR, MAPC, MAPCON, MAPLIST, MEMBER, MULT, NCONC, NOT, NULL, NUMBERP, ONEP, PAIR, PRIN1, PRINT, PUTSPACE, READ, RETURN, RPLACA, RPLACD, SASSOC, SASSQ, SET, STOP, SUB1, SUBST, SUM, TERPRI, TRACE, TRY, UNTRACE, WALK, UNWALK, ZEROP
FSUBR (19)	*FUNCTION, *PROGN, AND, ARRAY, COND, CSETQ, DEARRAY, ERRSET, FUNCTION, GO, LIST, LOCAL, OR, PROG, PROG2, PROGN, QUOTE, SETA, SETQ
APVAL (13)	*LF, *ESC, APVAL, EXPR, FEXPR, FSUBR, HEXPR, LABEL, LAMBDA, NIL, SUBR, T, HARRAY

3．結び　日頃御指導いただく関野治太郎教授並びに協力を仰いだ同研究室諸兄に感謝いたします。

参考文献[1] 井田：マイクロコンピュータを用いたLispマシンALPS/I；情報処理（投稿中）[2] 井田：記号処理研究委員会報告集P142-165, 1976
[3] 後藤英一：連載Lisp入門 bit 1974