数理解析研究所講究録
215巻 1974年 127-142

再帰的関数の評価機能

慶應義塾大学工学部　中西　正和

1. はじめに

再帰的に定義された関数を含む式の評価の方法は，LISP, SNOBOL, ALGOL その他の再帰的定義の可能な言語の処理系の製作技法の研究や，λ-calculus における S.E.C.D 機械の定義のような数学的な接近によって検討されている。その検討は，データ構造の設計の段階から処理アルゴリズムの改良まで広範囲にわたっている。

評価の方法の検討は，現存する機械の制約から，次の2つの方法で行なわれていると思われる。1つは再帰的関数の定義を配列を含む逐次処理型の手続き（流れ図の表現）に変えるための一般的なアルゴリズムを見つけていくことである。これはいわばコンパイラの製作技法の検討である。もう1つは，定義はそのままで，式の評価の方法を改良していく方法である。これはいわばインタープリタの製作技法の検討であ

る。S.E.C.D機械やLISPの万能関数に対する検討，またはこのためのデータ構造の考察などがこれにあたる。

ここで採っている立場は，インタープリタの検討である。コンパイラの技法も重要な研究課題であるが，動的な関数をコンパイルする場合や，非局所的変数のある関数をコンパイルすることを考えたとき，そこにはどうしても定義の原型の保持が必要になり，インタープリタの動作する部分が現われる。したがって，どのような場合にも翻訳機能（インタープリタ，ここではこれを評価機能と呼ぶ）の検討は必要であり無視することができない。

はじめに評価機能の形式的な定義を与え，次に改良された評価機能を提示する。さらにその機能の具体的な応用を検討する。そして最後に現実の処理系との比較と，インプリメントするときの1つのアイデアを提供する。

2.基本的な記号と定義

評価されるものを式という。式は一般に $a+b$ とか $f(x)-3\times g(x-1,y)$ のような形を持っているが，ここでは記法を統一するために，これらをすべて prefix form で表わす。評価機能はこの form にある操作を処す手続きである。

2.1 式の定義

予め定義されている機能の名前の集合を既定義関数（または端末関数）の集合と呼び，Tで表わす。Tは有限集合である。予め定義されている特殊機能の名前の集合を特殊既定義関数（または特殊端末関数）の集合と呼び，T_Eで表わす。既定義関数は一般の分野で＋や×などであり，既定義の演算である。特殊既定義関数は *if p then* e_1 *else* e_2 のような条件式のスケルトンや，変数の個数が定まっていないような関数，あるいは 'ABC' や *quote*[(A B)] のような引用演算などである。

一定の規則のもとに作られた名前の集合を名前の集合と呼び，Vで表わす。Vは有限の場合もあれば無限の場合もある。

予め値を定義されている名前の集合を定数の集合と呼び，Bで表わす。数を表わす10進表示の数字の列312や−13.3とか，LISPにおけるAPVAL定数などがこれにあたる。

Vの要素fに定義が与えられるとは，fに対応する入記号が存在することを言う。

i) Bの要素βは式である。

ii) Vの要素υは式である。

iii) $a_1, a_2, \cdots, a_n$ がそれぞれ式であり，t_n がTの要素でn個（$n \geq 0$）の引数を持つものならば $t_n(a_1, a_2, \cdots, a_n)$ は式である。

iv) $a_1, a_2, \cdots, a_n$ がそれぞれ式であり，f_nが定義を与えられたVの要素で対応するλ記号の変数部がn個の要素を持つならば（たとえば $\lambda x \lambda y \lambda z\ x+y+z = \lambda x,y,z\ x+y+z$ ならば $n=3$）$f_n(a_1, a_2, \cdots, a_n)$ は式である。

v) $a_1, a_2, \cdots, a_n$ がそれぞれ式であり，t_eがT_E の要素のとき，$t_e(L^{t_e}(\langle a_1, a_2, \cdots, a_n \rangle))$は式である。ただし $L^{t_e}(\langle a_1, a_2, \cdots, a_n \rangle)$ はt_eの定める文法に従って $a_1, a_2, \cdots a_n$ を配置した記号の列である。

vi) $a_1, a_2, \cdots, a_n$ がそれぞれ式であり，f_eが特殊定義を与えられたVの要素ならば $f_e(L^{f_e}(\langle a_1, a_2, \cdots, a_n \rangle)$ は式である。ただし $L^{f_e}(\langle a_1, a_2, \cdots, a_n \rangle)$ はf_eの定義による文法に従って $a_1, a_2, \cdots, a_n$を配置した記号の列である。

vii) i)～vi) で定義されたものだけを式と呼ぶ。

2.2 記号

$a_1, a_2, \cdots, a_n$のように a_i をコンマで区切って並べたものを<u>列</u>と呼び，$\boldsymbol{a}$と略記する。$n=0$ のとき空列と呼び，εと書く。$\langle \boldsymbol{a} \rangle$ を<u>リスト</u>と呼ぶ。$\langle \varepsilon \rangle$を<u>空リスト</u>と呼び，$\Lambda$と書く。$\boldsymbol{a} = a_1, a_2, \cdots, a_n$ で $\boldsymbol{b} = b_1, b_2, \cdots, b_m$ $(n \geq 0)$ のとき $a_1|b_1, a_2|b_2, \cdots, a_n|b_n$ を $\boldsymbol{a}|\boldsymbol{b}$ と書く。$\langle \boldsymbol{a}|\boldsymbol{b} \rangle$ を<u>組のリスト</u>と呼ぶ。

列 $\mathfrak{a}$ のなかに a という要素が存在するとき $a \prec \mathfrak{a}$ と書く。$\mathfrak{a}$ の中に a が存在しないとき $a \dashv \mathfrak{a}$ と書く。

$a = \langle a_1, a_2, \cdots, a_n \rangle$ $(n \geq 1)$ とする。$'a \equiv a_1$ であり，$a' \equiv \langle a_2, \cdots, a_n \rangle$ である。また，任意のリスト $a = \langle a_1, a_2, \cdots, a_n \rangle$ $(n \geq 0)$ に対し，$\pi(b, a) \equiv \langle b, a_1, a_2, \cdots, a_n \rangle$ である。

a が空リストのとき $a \cdot b \equiv b$，a が空でないリスト，b が任意のリストのとき $a \cdot b \equiv \pi('a, a' \cdot b)$ である。この演算 $\cdot$ を<u>リストの接続</u>と呼ぶ。

<u>組のリストの差</u> $a - b$ を次のように定める。a が空リストならば $a - b \equiv \Lambda$。$\boldsymbol{x} = x_1, x_2, \cdots, x_n$，$a = \langle \boldsymbol{x} | \mathfrak{a} \rangle$，$b = \langle \boldsymbol{y} | \mathfrak{b} \rangle$ とする。$x_1 \prec \boldsymbol{y}$ ならば $a - b \equiv b'$。$x_1 \dashv \boldsymbol{y}$ ならば $a - b \equiv \pi('a, a' - b)$。<u>組のリストの和</u> $+$ は $a + b \equiv a \cdot (b - a)$ で定義する。

$f(x_1, \boldsymbol{\ell}), f(x_2, \boldsymbol{\ell}), \cdots, f(x_n, \boldsymbol{\ell})$ のような列を $\boldsymbol{f}^{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\ell})$ と略記する。混同の恐れがないときは単に $\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\ell})$ と書く。以下では，$\boldsymbol{f}(\boldsymbol{x}, \boldsymbol{\ell})$ と書いたときは特にことわらない限り $\boldsymbol{f}^{\boldsymbol{x}}(\boldsymbol{x}, \boldsymbol{\ell})$ であるとする。

$x \prec \boldsymbol{x}$ とする。$x = '\langle \boldsymbol{x} \rangle$ ならば $\xi(x, \langle \boldsymbol{x} | \mathfrak{a} \rangle) \equiv '\langle \mathfrak{a} \rangle$。$x \neq '\langle \boldsymbol{x} \rangle$ ならば $\xi(x, \langle \boldsymbol{x} | \mathfrak{a} \rangle) \equiv \xi(x, \langle \boldsymbol{x} | \mathfrak{a} \rangle')$。

α が T または T_E の要素のとき，$\gamma(\alpha)$ は既定義関数名また

特殊既定義関数αに対応する機能を表わす。また，βがBの要素ならば$\gamma(\beta)$は定数βの値を表わすものとする。

fがVの要素で，定義が与えられているとき，$\mu(f)$は対応するλ記号の変数部の逆順のリストであり，$\delta(f)$は本体部である。すなわち，fが$\lambda x_1(\lambda x_2 \cdots (\lambda_n E)\cdots)$の定義を与えられているならば，$\mu(f) \equiv x_n, \cdots, x_2, x_1$であり，$\delta(f) \equiv E$である。$x = x_1, x_2, \cdots, x_n$ならば$rev(x) \equiv x_n, \cdots, x_2, x_1$である。

3. 評価機能の定義

評価機能は1つの手続きとして定義する。前節で定義した・やξなどの演算は1つの既定義手続きを表わすものとする。定義のなかで使われる記号は次のように定める。$\beta \in B$，$v \in V$，$\alpha \in T$，$f \in V$かつfに対応するλ記号が存在する。$\alpha_E \in T_E$，$f_E \in V$かつf_Eに対応するλ記号が存在し，特殊関数であることが宣言されている。

次のように定義するE_Vを評価機能と呼ぶ。

$$E_V(e) \longrightarrow E(e, \Lambda)$$

i) $E(\beta, \langle x|e\rangle) \longrightarrow \gamma(\beta)$

ii) $E(v, \langle x|e\rangle) \longrightarrow \xi(v, \langle x|e\rangle)$

iii) $E(\alpha[a], \langle x|e\rangle) \longrightarrow \gamma(\alpha)(E(a, \langle x|e\rangle), \langle x|e\rangle)$

iv) $E(f[a], \langle x/e\rangle) \rightarrow E(\delta(f), \langle \mu(f) / rev(E(a, \langle x/a\rangle))\rangle \cdot \langle x/e\rangle)$

v) $E(\alpha_E[L^{\alpha_E}(\langle a\rangle)], \langle x/e\rangle) \rightarrow \gamma(\alpha_E)(\langle L^{\alpha_E}(\langle a\rangle)\rangle, \langle x/a\rangle)$

vi) $E(f_E[L^{f_E}(\langle a\rangle)], \langle x/e\rangle) \rightarrow E(\delta(f_E), \langle \mu(f_E) / \langle x/e\rangle, \langle L^{f_E}(\langle a\rangle)\rangle\rangle \cdot \langle x/e\rangle)$

ここでvi)の定義から明らかなように$\mu(f_E)$は2つの要素を持つ列である。$\mu(f_E) = \mu_1, \mu_2$の意味づけは次の通りである。μ_1はこの時点での評価の環境（組のリスト）のための変数であり，μ_2は$\langle L^{f_E}(\langle a\rangle)\rangle$すなわち$f_E$の持つ文法に従う記号の列の1まとまりのための変数である。したがってf_Eには2変数の関数の定義が与えられなければならない。

4. 条件式

条件式の評価のための機能はEにおいてはT_Eに属する特殊既定義関数の一種である。この関数はよく使われるものであるので，ここにその定義を述べる。特殊既定義関数名をCとする。その文法は$Cd[p_1 \rightarrow e_1, p_2 \rightarrow e_2, \cdots, p_m \rightarrow e_n]$とする。ここで$p_i$はその値が真または偽となる式であり，$e_i$は$p_i$が真となったとき全体の値となるべき値をとる式である。

Cdの定義を次のような入記号で与える。この定義の中で，

pl は $pl(a \to b) = a$ となるような手続きであり，pr は $pr(a \to b) = b$ となるような手続きである。

$$\gamma(Cd) = \lambda e,m\,(\text{if } E(pl('e), m) \text{ then } E(pr('e), m) \text{ else } \gamma(Cd)(e', m))$$

または $\gamma(Cd)$ を C と表わして

$$C(a, \langle x|e\rangle) \to \text{if } E(pl('a), \langle x|e\rangle) \text{ then } E(pr('a), \langle x|e\rangle) \text{ else } C(a', \langle x|e\rangle)。$$

このように特殊既定義関数ではその与えられた記号列の中に存在する式が既定義関数の定義の中に現われている E によってはじめて評価される場合がある。また，引用関数などでは評価が行なわれない。

5. 組リスト抑制型の関数

F_V を次のように定義する。$\beta, \upsilon, f, \alpha$ は前と同じである。

$$F_V(e) \to F(e, \Lambda, \Lambda)$$

i) $F(\beta, \langle x_L|e_L\rangle, \langle x_N|e_N\rangle) \to \gamma(\beta)$

ii) $F(\upsilon, \langle x_L|e_L\rangle, \langle x_N|e_N\rangle) \to \xi(\upsilon; \langle x_N|e_N\rangle \cdot \langle x_L|e_L\rangle)$

iii) $F(\alpha[a], \langle x_L|e_L\rangle, \langle x_N|e_N\rangle) \to \gamma(\alpha)(F(a, \langle x_N|e_N\rangle \cdot \langle x_L|e_L\rangle, \Lambda), \langle x_N|e_N\rangle \cdot \langle x_L|e_L\rangle)$

iv) $F(f[a], \langle x_L|e_L\rangle, \langle x_N|e_N\rangle)$

$\to F(\delta(f), \langle x_L|e_L\rangle, \langle \mu(f)|new(F(a, \langle x_N|e_N\rangle \cdot$

$\langle x_L/e_L\rangle, \Lambda))\rangle + \langle x_N/e_N\rangle)$

v) $F(\alpha_F[L^{\alpha_F}(\langle a\rangle)], \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

$\rightarrow \gamma(\alpha_F)(\langle L^{\alpha_F}(\langle a\rangle)\rangle, \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

vi) $F(f_F[L^{f_F}(\langle a\rangle)], \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

$\rightarrow F(\delta(f_F), \langle x_L/e_L\rangle, \langle \mu(f)|\langle x_N/e_N\rangle \cdot \langle x_L/e_L\rangle,$

$\langle L^{f_F}(\langle a\rangle)\rangle\rangle + \langle x_N/e_N\rangle)$

α_Fおよびf_Fはそれぞれα_E, f_Eに対応するもので，3つの変数を持つ機能で定義される．たとえば前節での条件式評価関数Cは次のようなDになる。

$D(a, \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

$\rightarrow$ if $F(pl('a), \langle x_N/e_N\rangle \cdot \langle x_L/e_L\rangle), \Lambda)$

then $F(pr('a), \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

else $D(a', \langle x_L/e_L\rangle, \langle x_N/e_N\rangle)$

一般にα_Fおよびf_Fにおける Fの参照部分に関して次のような置きかえ規則を設定する。

まず，T_Eのどの要素をも，すべて次のような定義を持つとしよう。

$\gamma(te)(e, m) \rightarrow$ if p_1 then a_1 else

if p_2 then a_2 else

---

if p_n then a_n $\quad (n \geq 0)$

もちろん，$\gamma(te)=a_1$ のような定義も含む（これは $n=1$ で p_1 が真だけをとる命題であるとみなす）．

まず，定義の左辺 $\gamma(te)(e, m)$ を $\gamma(t_f)(e, m_L, m_N)$ でおきかえる。次に右辺の a_i の形が $E(\rho(e), m)$ の形ならばこれを $F(\rho(e), m_L, m_N)$ でおきかえる。$ue \in T_E$ のとき，a_i が $\gamma(ue)(\rho(e), m)$ の形のとき これを $\gamma(uf)(\rho(e), m_L, m_N)$ におきかえる．ρ は $\rho(e)=e$ なるものも含む．

そのほかの ue や m の現われるところは（p_i の中も含む）それぞれ次のように置きかえる。$\gamma(ue)(a, b) \longrightarrow \gamma(uf)(a, b, 1)$. $E(a, b) \rightarrow F(a, b, 1)$. $m \longrightarrow m_N \cdot m_L$. f_F についても同様である。このようにして作られた t_f の集合を T_F と書く。

6. E_V と F_V に関する考察

前節で述べた te に関し，次のような仮定を置く。T_E の要素に与えられる定義の中に現われる E は，すべて $E(a, u(m))$, $E(a, m)$, $E(a, s \cdot m)$ のどれかの形で現われる．u は $u(l \cdot (m_1+m_2) \cdot n)=u(l \cdot m_1 \cdot m_2 \cdot n)$ である。また E の参照を含まない φ で $\varphi(l \cdot (m_1+m_2) \cdot n)=\varphi(l \cdot m_1 \cdot m_2 \cdot n)$. m および φ のパラメタは与えられた te の定義の第2引数である．

E や F の定義の中の矢印は，左辺のものが右辺に置きかわることを示している．何回かの置きかえの後，a が b になり，

kのおきかえがもうできないとき$a \Rightarrow b$と書く．$f(a) \Rightarrow e$でかつ$g(b) \Rightarrow e$のとき$f(a) = g(b)$と書く。また，$a \Rightarrow b$のとき，適用された関数fの適用回数を$N_f a \Rightarrow b$ または，$N_f a$と書く。また$\gamma(\alpha)(s, l \cdot m_1 \cdot m_2 \cdot n) = \gamma(\alpha)(s, l \cdot (m_1 + m_2) \cdot n)$であると仮定する．

補助定理1．

$$\xi(v, l_1 \cdot l_2) = \xi(v, l_1 + l_2)$$

証明略

補助定理2．

$$E(e, l \cdot m_1 \cdot m_2 \cdot n) = E(e, l \cdot (m_1 + m_2) \cdot n) \quad \text{かつ}$$

$$N_E E(e, l \cdot m_1 \cdot m_2 \cdot n) = N_E E(e, l \cdot (m_1 + m_2) \cdot n)$$

証明

$N_E(E(e, l \cdot m_1 \cdot m_2 \cdot n)) = 1$ならば$e$は$\beta$または$v$の形に限られる。$e$が$\beta$ならば明らかである．$e$が$v$の形ならば$E(v, l \cdot m_1 \cdot m_2 \cdot n) = \xi(v, l \cdot m_1 \cdot m_2 \cdot n)$．補助定理1より$E(v, l \cdot (m_1 + m_2) \cdot n) = \xi(v, l \cdot (m_1 + m_2) \cdot n) = \xi(v, l \cdot (m_1 \cdot m_2) \cdot n) = \xi(v, l \cdot m_1 \cdot m_2 \cdot n)$．$k > 1$のとき，$N_E E(e, l \cdot m_1 \cdot m_2 \cdot n) = m < k$なる$e, l, m_1, m_2, n$に対して$N_E E(e, l \cdot (m_1 + m_2) \cdot n) = m$でかつ$E(e, l \cdot m_1 \cdot m_2 \cdot n) = E(e, l \cdot (m_1 + m_2) \cdot n)$であると仮定する．$N_E E(e, l \cdot m_1 \cdot m_2 \cdot n) = k$なる$e, l, m_1, m_2, n$を考える。$k > 1$であるから$e$は$\alpha[a]$, $f[a]$, $\alpha_E[L^{\alpha_E}($

$\langle a \rangle)]$ または $f_E[L^{f_E}(\langle a \rangle)]$ のどれかの形である。e が $\alpha[a]$ の形ならば $E(\alpha[a], l \cdot m_1 \cdot m_2 \cdot n) = \gamma(\alpha)(E(a, l \cdot m_1 \cdot m_2 \cdot n), l \cdot m_1 \cdot m_2 \cdot n)$。帰納法の仮定から $E(a, l \cdot m_1 \cdot m_2 \cdot n) = E(a, l \cdot (m_1 + m_2) \cdot n)$ かつ $a_i \prec a$ に対し $N_E E(a_i, l \cdot m_1 \cdot m_2 \cdot n) = N_E E(a_i, l \cdot (m_1 + m_2) \cdot n)$。$\gamma(\alpha)$ の仮定から明らかである。e が $f[a]$ ならば $E(f[a], l \cdot m_1 \cdot m_2 \cdot n) = E(\delta(f), \langle \mu(f) | rev(E(a, l \cdot m_1 \cdot m_2 \cdot n)) \rangle \cdot l \cdot m_1 \cdot m_2 \cdot n) = E(\delta(f), \langle \mu(f) | rev(E(a, l \cdot (m_1 + m_2) \cdot n)) \rangle \cdot l \cdot m_1 \cdot m_2 \cdot n) = E(\delta(f), (\langle \mu(f) | rev(E(a, l \cdot (m_1 + m_2) \cdot n)) \rangle \cdot l) \cdot (m_1 + m_2) \cdot n) = E(e, l \cdot (m_1 + m_2) \cdot n)$。また、$N_E E(\alpha[a], l \cdot (m_1 + m_2) \cdot n) = k$, $N_E E(f[a], l \cdot (m_1 + m_2) \cdot n) \doteqdot k$ は明らかである。e が $\alpha_E[L^{\alpha_E}(\langle a \rangle)]$ のとき $E(\alpha_E[L^{\alpha_E}(\langle a \rangle)], l \cdot m_1 \cdot m_2 \cdot n) = \gamma(\alpha_E)(\langle L^{\alpha_E}(\langle a \rangle) \rangle, l \cdot m_1 \cdot m_2 \cdot n)$。$\gamma(te)$ の仮定から、$\gamma(\alpha_E)$ の定義の中に現われる E はすべて $E(b, u(m))$, $E(b, m)$ または $E(b, s \cdot m)$ のどれかの形である。また、定義の中での E の参数を含まない関数 φ はどれも $\varphi(l \cdot m_1 \cdot m_2 \cdot n) = \varphi(l \cdot (m_1 + m_2) \cdot n)$ であるから明らかである。f_E も t_E と同様の仮定のもとに証明される。

<u>定理</u>

$E(e, l_1 \cdot l_2) = F(e, l_2, l_1)$ かつ $N_E E(e, l_1 \cdot l_2) = N_F F(e, l_2, l_1)$

証明

β, vについては省略する. $e=\alpha[a]$ のとき $E(\alpha[a], l_1\cdot l_2)=\gamma(\alpha)(E(a, l_1\cdot l_2), l_1\cdot l_2)$. 一方 $F(\alpha[a], l_2, l_1)=\gamma(\alpha)(F(a, l_1\cdot l_2, \Lambda), l_1\cdot l_2)$. 帰納法の仮定より $a\prec a$ に対し $E(a, \Lambda\cdot l_1\cdot l_2)=F(a, l_1\cdot l_2, \Lambda)$. したがって $E(\alpha[a], l_1\cdot l_2)=F(\alpha[a], l_2, l_1)$. $e=f[a]$ ならば, $E(f[a], l_1\cdot l_2)=E(\delta(f), \langle\mu(f)|rev(E(a, l_1\cdot l_2))\rangle\cdot l_1\cdot l_2)$. $F(f[a], l_2, l_1)=F(\delta(f), l_2, \langle\mu(f)|rev(F(a, l_1\cdot l_2, \Lambda))\rangle+l_1)$. $a\prec a$ なる a に対し $E(a, l_1\cdot l_2)=F(a, l_1\cdot l_2, \Lambda)$. したがって $E(e', l\cdot l_1\cdot l_2)=F(e', l_2, l+l_1)$ を示せばよい(このとき $N_E E(e', l\cdot l_1\cdot l_2)<k$ である). 帰納法の仮定より $F(e', l_2, l+l_1)=E(e', (l+l_1)\cdot l_2)$. 補助定理2より $E(e', (l+l_1)\cdot l_2)=E(e, l\cdot l_1\cdot l_2)$. $e=\alpha_E[L^{\alpha_E}(\langle a\rangle)]$ ならば $t_e\to t_f$ の置きかえにより $\alpha[a]$, $f[a]$ の場合と同様になる. f_Eも同様である.

系

$E_V(e)=F_V(e)$

証明

定理における $l_1=l_2=\Lambda$ なる場合である.

7. インプリメンテーションに関する考察

Fvを単にその定義に従ってインプリメントするならば，パラメータが1つ増えただけ損であり何のメリットもない。ところがFvには次のような特徴がある。

i) Fの逐次的定義の部分（f[a]の部分）では，その組リストの大きさが必要最小の大きさに制限される。

ii) Fの相互再帰的定義の部分では，組リストの破壊がないように保護される（十の演算により削減することはない）。

すなわち，Eではすべてを保護しているのに比べ，Fでは必要なものだけを保護している。このことは次のような結果をもたらす。たとえばf[x,y]＝if x=0 then y else f[x-1, y+x]のような定義のfを使ってf[x,y]を評価すると，組リストの長さはEのとき最大2x+2であり，Fのときは2である。

LISP1.5を例にとると，$\langle x_N | e_N \rangle$の部分（Fの才3の引数）は翻訳の途中で臨時に発生したものと考えることができるので，高速のレジスタの集まり（たとえば連想記憶装置）に記憶することができる。$\langle x_L | e_L \rangle$は従来通りのa-listである。この結果，かなり速い処理系を作ることができる。さらにa-listの延長が少なくなるためにスペース効率が良く，ミニコンによるLISPも実用性を持ち得ることになろう。

Fの欠点は次のようなものである。すなわち，T_Eの要素，

Tの要素に対して，Fを作るためにいくつかの仮定を設けてある。このことはLISPではFEXPRに対する制限となって現われるが，一般に使うことは皆無であろうと思われる事ばかりと信じている。たとえばa-listを値とする関数を定義することなどであるが，これはその関数の引用の仕方が，たとえばsassocのようにリストのはじめから捜すような場合には何の支障もない。支障が起こるのは，a-listの3番目の要素は何かといった場合で，このようなときはEとFの結果が異なる。しかしこのような特殊な場合は皆無といってよいだろうし，またそのような使い方はEの中に利用者が手を入れる場合と考えても良いのだから大きな欠点とは思われない。もう1つはLISPの$ALISTのようなa-listの静的な参照がしづらいことであるが，これも同様の欠点と思われる。

現在PDP-11/21（8KW）に，この方式によるLISPを組み入れるべく進行中である。

8.謝辞

現在このアイデアを進めることのできるのは筑波大学の西村敏男先生の多大の御指導と励ましのおかげである。深く感謝する次第である。立教大学の島内剛一先生，早稲田大学の広瀬健先生，東京工業大学の木村泉先生には貴重な御教示を

いただいた・深く感謝する次第である.