

## L<sup>6</sup> についての覚え書き

統計数理研 渋谷 政昭

1)

L<sup>6</sup> はベル研究所の K.C. Knowlton が動画作製用の計算機言語として開発したもので、名前は laboratory's low level linked list language の6つのlによる。この言語の最大の特徴は低水準といえる。 (リスト処理言語ではどのような特徴を高水準と呼ぶべきか必ずしもはっきりしないが。) Knowlton は IBM 7094, GE 635 で L<sup>6</sup> を開発した。日東では HIPAC-103 (統教研), TOSBAC 3400<sup>-TOPS X11</sup> (日東科学技術研修所), FACOM 270, 20/30 (東大理, 後藤研) で implement されている。後者2つは近日公開されるであろう。

2)

L<sup>6</sup> の4つの基本実体とその特徴は次の通りである;  
7口77. サイズが可変であり 2<sup>n</sup> 論理語 (n=0, 1, ..., 7) のものを使用できる。

区分. ブロックの, 2語にわたる任意の位置に, 任意の長さの区分を設け, これに数, 文字, 等のデータを記憶できる. 区分の定義はすべてこのブロックに共通であり, 1文字 (A~Z, 0~9) の名前が付けられる. 区分の定義は動的に (計算実行中に計算結果に依存して) 行なうこともできる.

指示子. 何らかの方法でブロックを指定するデータ (たとえば「ブロックの先頭の構文語の番地) を指示子という. 指示子は一定長以上の長さの任意の区分に記憶できる. システムより眺めると, 指示子を他のデータと区別することはできない.

基地レジスター. 1論理語の大きさの記憶で指示子を格納するが作業番地としても用いられる. 使用目的によ, これは出と呼ぶことがふさわしい. これも1字 (A~Z) の名前をもっている.

連結リストとは指示子により互に結びつけられたブロックと基地レジスターの集合のことである. あるブロックのある区分について演算する (区分の内容をとり出し, それについて演算し, とくに格納する) には, 基地レジスターより出発して指示子の入っている区分の名前を順次に羅列することにより必要なブロックにたどりつき, 最後に区分の名前を書くことにより指定できる. 例. BPPD: 基地レジスター B にある指示子の指示するブロックの・区分 P にある指示子の指示

するブロックの・区画 P にある指示子の指示するブロックの・区画 D.

区画名を1字に限ることにより, 離れたブロックにある区画を短い語で指示でき, しかもある程度は記憶の便となる.

3)

L<sup>6</sup> の言語の単位は文 (FORTRAN 同様に原則として1行) であり次り形をしている.

a) 無条件文.

<名札> THEN <基本演算> . . . <名札>

<基本演算> の典型的な形は (<区画>, <演算コード>, <区画> または <リテラル>) であるが, 引数の数がより多くなるものもある. 文末に <名札> があれば無条件飛越し, なければ次行の文に移る.

b) 条件文.

<名札> <ifs> <基本判別> . . . THEN <基本演算> . . . <名札>

<基本判別> は <基本演算> と同じ形をしており, 区画の内容を他の区画の内容またはリテラルと比較し, <真> または <偽> の値をとる. <ifs> は IFALL (または IF), IFANY, IFNONE (または NOT), IFNALL のいずれかで, 二つまたは  $x_i$  を基本判別の論理値とすると, それぞれ  $\cap x_i$ ,  $\cup x_i$ ,  $\cap \bar{x}_i$ ,  $\cup \bar{x}_i$  を意味する. このいずれの形式のためプロ

グラフは比較の見易い。

4)

自由ブロック・リストからのブロックの獲得 (get) の基本演算は ( $\langle \text{区分} \rangle$ , GT,  $\langle \text{区分} \rangle$  または  $\langle \text{整数} \rangle$ ) である。ホ3引数の指定する大きさのブロックを取って、それへの指し子をホ1引数の区分に入れる。基本演算は ( $\langle \text{区分} \rangle$ , GT,  $\langle \text{区分} \rangle$  または  $\langle \text{整数} \rangle$ ,  $\langle \text{区分} \rangle$ ) とホ4引数をもつときには、ホ1引数の旧い内容がホ4引数の区分に入れられる。基地レジスタ  $R$  より出発して区分の指し子でつらなれれているブロックの単純リスト (冊) があるとき ( $R$ , GT,  $\langle \cdot \rangle$ , RL) は冊の押し下げとす。

逆にブロックの自由化 (free) は ( $\langle \text{区分} \rangle$ , FR, 0) で行なう。ホ1引数の区分が指し示すブロックを自由ブロックに戻し、区分に 0 を入れる。もしも ( $\langle \text{区分} \rangle$ , FR,  $\langle \text{区分} \rangle$ ) ならば、<sup>自由化の後</sup>ホ3引数の区分の内容がホ1引数の区分に移される。上の冊の例で ( $R$ , FR, RL) は逆に跳ね上げとす。ブロックの自由化は完全に使用者の責任であり、システムはごみ集めをしない (できずい)。

5)

サブルーティン呼出しは ( $\text{DO}$ ,  $\langle \text{名札} \rangle$ ) と...; 基本演算で行なう。この基本演算の次にある基本演算への戻り着地が

システムのもつ棚にのせられ、〈名札〉の<sup>文</sup>棚に命令の流水が移る。FORTRAN の RETURN に相当する DONE と…) 特殊の〈名札〉があり、これが文末にあると、棚の一番上の床り着地を踏み上げ、そこは命令の流水を移す。カプラーは、FORTRAN, ALGOL のようには、まじりとした単位でけり、上のように DO, DONE で制御が行われるようになってある。したがって帰納的を吟出しも可能である。

パラメータの授受は使用者の構成する棚によつて行われ、このよいか、補助手段としてシステムが区分の内容 (field content) の棚をもっている。そこは載せ (save), 下す (restore) する基本演算は (S, FC, 〈区分〉), (R, FC, 〈区分〉) である。カプラーテンを一般化するために、1つ区別の定義の棚もある。

6)

L<sup>6</sup> は何れかであるために、諸現象・対象のもつてい構造をそのまま記憶で表現できる。たとえば算術式を、節に演算記号をもった2進樹木として表わされる。(LISP 7222 はすべてを葉先に移すために余分のブロックを要する。) しかもその際に若干の考察により記憶を節約し、計算速度を早めるよう区別の定義を行ふことが出来る。

対称リスト、環構造 (ring structure) なども容易に構成



第10回プログラミング・シンポジウム。

日科技術計算センター (1970) : T-L<sup>6</sup>。使用者のための  
便覧。

—— (1970) : T-L<sup>6</sup> によるリスト処理入門。