

---

 寄 書
 

---

## ヨ - ロ ッ パ の ALGOL\*

井 上 謙 藏\*\*

今年の3月16日から19日にかけてミュンヘンでIFIPのWG 2-1が行なわれた。これはIFIPのTechnical Committee 2の作業グループでALGOLの研究を進めている組織である。日本からは森口教授(東大工学部)が委員となっておられるわけであるが、今度会議には多忙のため、わたくしが代理として出席することになった。そこでの討論や結論を伝えるのがわたくしの任務かも知れないけれども、不正確な伝達の仕方をするよりは、それは何れ発行されるはずの印刷物にゆだねることにして、会議の様子や、その前後におとずれた2, 3の計算センタの様子から、ヨーロッパのALGOL implementorたちの動きをお伝えしたい。

## 1. WG 2-1 の第三回会議

前のデルフト(Delft)会議に続いて、今度は第三回目、本来はミュンヘンのHochschuleで開かれることになっていた。ところが同じ頃、建築機械の学会と展示会がミュンヘンにあったので、WG 2-1の方はそこから汽車で南方、約40分のトゥッツィング(Tutzing)という小さな町のホテルに確詰となることになった。湖のほとりの大変景色のよい所だそうであるが、連日雪と雨で、そのほうはさっぱりお目にかかることができなかった。

さて、あらかじめ知らされた議題は

1. ALGOL のサブ・セット
2. ALGOL の入出力
3. Future ALGOL

ということで、1. についてはすでにデルフト会議に電気試験所の和田部長が出席される直前から討論されていたので問題の所在をよく知っていたし、入出力に

ついても出発の前に、ACMのプログラム言語委員会、今年はじめにISOに出した論文を入手したので大体のことはわかっている心算であった。Future ALGOLに関しては

Niklaus Wirth: A Generalization of ALGOL.

C. ACM. 547, No. 9, Vol. 6, 1963.

という論文がある。不勉強にして読んでいなかったので飛行機の中に持ちこんだが結局眺めただけになってしまった。その底にはfuture ALGOLというのは本当にfutureの問題で、せいぜい問題の所在をさぐる討論が行なわれるにすぎないだろうという下心がはたらいっていたわけである。会議の進行につれてこれは完全に裏切られてしまった。時間比でいうと、午前と午後、計10のsessionのうち、サブ・セットとI/Oの討論に費やしたのは3 session ぶんぐらいであり、あとは全部future ALGOLの講義と討論である。

サブ・セットについては、すでに知られているとおりで殆んど変更はない。入出力に関しては

Report from WG 2-1 on Input-Output Procedures.

というのが出されることになった。このレポートの精神は、ALGOL 60 というものが大変柔軟性をもつ言語であるから必要最小限の入出力手続きをきめておけば、あとは利用者が勝手に自分の都合のよい入出力手続きを組み立てることができるということである。必要最小限というのはどうしてもコードで書かねばならない部分ということで、それは、

in symbol, out symbol, length, in real,  
out real, in array, out array

の7個のprocedureであるin symbolの宣言は

procedure in symbol (channel, string, destination);

value channel; integer channel, destination;

\* The Thoughts of the European ALGOL Implementors, by Kenzo INOUE (The University of Tokyo).

\*\* 東京大学物性研究所

**string string; <Procedure body>**

で、これはたとえば **string** が abc であって、テープ読取器から読んだ **character** が a であれば整数 1 を、b であれば整数 2 を、c であれば整数 3 を、a, b, c 以外の **basic symbol** なら 0 を 整数型変数 **destination** に **assign** する **procedure** である。たとえばこれで整数読込の手続きを作ると、

```

procedure in integer (channel, integer);
  value channel; integer channel, integer;
  begin integer n, k; Boolean b;
    integer := 0; b := true;
    for k:=1, k+1 while n=0 do
      insymbol(channel, '0123456789-+',
        n);
    if n=11 then b := false else if n=12 then
      else integer := n-1;
    for k:=1, k+1 while n≠13 do
      begin integer := 10 × integer + n-1;
        insymbol (channel, '0123456789-
          +', n)
        end 1;
    if -b then integer := -integer
  end

```

という大変めんどろなことになる。length というのは **parameter** に与えられた **string** の長さを求める **integer procedure** である。要するに ALGOL 60 の柔軟性をできるだけ損ねたくないというのであろうが、その結果として、さきに述べた ACM の委員会の論文

#### A proposal for Input-Output Conventions in ALGOL 60.

を参照してくれということ結論にかかざるを得ないことになる。これと委員会の報告とは、その手続きの便利さにおいて月とすっぽん程の相違があり、日本の ALGOL compiler でも、たいしてはこれより便利になっているはずである。

Future ALGOL に関する議論には、全く聞いたこともない話がでてきた。ALGOL 6 X というのは知っていたが、どうも WG 2-1 のメンバーは、6 X には一寸立ち止まってみただけで、6 Y の方に関心が向いてしまったらしい。6 X というのは 60 をサブ・セットとして含むのであるが、6 Y は精神が 60 と似ているだけらしい。演説の皮切りは Wijngaarden がやっ

たし、彼の印刷物はちゃんと言葉が書いてあるので少しは内容を理解できた。

われわれは ALGOL 60 を主として数値解析のためのプログラム用言語と考えているが、WG 2-1 ではそれよりも数値解析のための **descriptive language** としての面を重視している。もし WG 2-1 が入出力手続きに対して低い関心しかもたないという判断が正しいとすれば、それは上記の立場に基くのであろう。ALGOL 6 Y はこの **descriptive language** としての性格を、数値解析としての範囲をこえて徹底的におしすすめようとする。

話は、「一般的な言語を作るときに文法的規定をなるべく少なくすべきだ」ということではじまる。つまり、一般性を制限するのが文法なのであるから、文法などはない方がよいというわけである。しかし何もなかったら何もできないから、いわば公理系だけをきめよう。たとえば、あるプログラムを処理する際に  $X+Y$  という式がでてきたとする。そのとき processor は、すでに処理のおわったプログラムの部分に  $X+Y$  の意味を規定する記述があるかどうかを **search** するわけである。この手続きを **value{X+Y}** とかく。いま ALGOL 60 の文法的規定がプログラムの中に含まれているとすれば、

$$\text{value}\{\langle\text{sum}\rangle+\langle\text{term}\rangle\}=\text{value}\{\text{value}\langle\text{sum}\rangle+\text{value}\langle\text{term}\rangle\}$$

なる記述を、(Wijngaarden の表現法で) みつけることができる。そこで X や Y が、この  $\langle\text{sum}\rangle$  や  $\langle\text{term}\rangle$  であれば、 $X+Y$  の **value** をとることはそれぞれの X, Y の **value** の和の **value** をとることになるわけである。そこで X に関する記述を、すでに読まれたプログラムの部分の中に **search** すると

**X in letter**

というのがある。次に

**letter in identifier**

をみつける。このようにして、ついに

**factor in term, term in sum**

と進む。すなわち

**X in sum**

である。Y についても同様だから、

$$\text{value}\{X+Y\}=\text{value}\{\text{value}\{X\}+\text{value}\{Y\}\}$$

である。さらに **search** をくり返して

**X=Z, Z=3**

をみつけたとすると

**value X = value 3**

となる。3に対して search をくり返すと3の value を定める式をみつけることができない。そのときは3自体で意味を完結しているものとする。

value X=3

Yについても

value Y=4

になったとする。そこで

value(X+Y)=value{3+4}

3+4 についてはさらにこれが7になるという記述がすでに読まれたプログラムの部分になければならない。もしこれがあれば

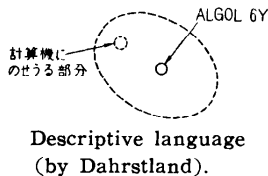
value(X+Y)=value 7=7

である。ここで value, in, <, >, [, ], =, alphabet, 3, 4, 7, +, - などは basic symbol か delimiter であるが、これら基本的な概念として、最小限何をとるべきか、Wijngaarden にもまだはっきりしていないらしい。あるところでは + や - を delimiter からはずし、+ や - を使いたければその意味を定義しなければならないといっている。そしてどんな文法を含んだプログラムがきても処理できるための言語の公理系としては何をもちねばならないかという議論に進む。

こんな考え方であると、ALGOL 60でも、ALGOL 6Xでも、ALGOL 6Yでも、あるいは今後考えるいかなる文法の言語でも、それで書いたプログラムの中に basic symbol の意味にまで下っていくことのできる文法を含むならば、それは必ず処理することができることになる。大変雲をつかむような話なので Facit の Mr. Dahrstland に、「こんな言語のプロセッサをわれわれは作ることができるであろうか」と質問してみた。彼は「いま考えつつあるのは将来展開されるはずの descriptive language の Kernel にすぎない。そしてわれわれは、そのごく一部分を計算機にのせうのみで

あろう」と、図をかいて甚だ楽しげに笑っていた。何だかギリシヤの哲人と立ち並ぶ大理石の円柱の間を逍遙しているような気の遠くなる話である。

会議の雰囲気も内容にふさわしく、Naur はビールをのみながら、Dijkstra は歩きまわりながら、そして Seegmüller は机の上に足をのせて、熱心に討議



を進める。Van der Poel はたえずお菓子をたべながら、しかも討論の紛糾を実にあざやかにさばくという神業をやっている名議長である。Future ALGOL のリーダー・シップは Wijngaarden がもち、Dijkstra, Naur, Wirth などそれに続いているらしい。

## 2. 計算センタ

WG 2-1 の前に Naur の Regnecentralen (コペンハーゲン) と、Wijngaarden のいる Mathematisch Centrum (アムステルダム) をおとずれた。

Naur は、彼の計算センタで develop した GIER と称する機械のコンパイラを 1963 年春に完成しており、その内容を得々と説明してくれた。GIER 自身は、磁心記憶 1,024 語、磁気ドラム 12,800 語 (40 語 × 3,200), 1 語 40+2 (flag) 桁, 1 命令/語または 2 命令/語の簡単な機械であるが、速度は加算固定 50 μs, 浮動 100 μs とかなり速い。外見もデコラの木目模様の、居間にもおけそうなスマートなデザインで、50 台ぐらい売れているそうである。GIER ALGOL は ALGOL 60 の文法に次の制限を加えたものである。

1. 値で呼ばれる label, array の formal parameter は許されない。
2. procedure のパラメータはすべて specify する。
3. 整数の label, own array はない。

したがって recursive procedure は許されるのであって、実に嬉しそうに、これを実験して見せてくれた。方法は、Dijkstra の考え方とほとんど同じで、object program の run time に徹底して stack を用いるのである。彼は、この方法で object run の能率がおちることはないと主張している。コア記憶が少ないので object program の 40 語を 1 segment とし、ドラムにおき、最もひんぱんに使われる segment をできるだけ多くコアにもってきおく。ATLAS のページ式記憶の administration に刺激されたらしい。1階が映画館のビルの2階、3階を占めた明るい、若者ばかりの活気にみちた計算センタであった。そして Naur は頭の回転の早い、茶目けたっぶりのヒゲ男である。

アムステルダムの計算センタは、コペンハーゲンのそのスマートさはない。Dijkstra もすでに去っている。計算機は Electrosica, 全くのバラック・セットで電線が床をはいまわっており、紙テープがちらか

っている。速度は固定加算で 64  $\mu$ s, 磁心記憶 12,000 語, 固定記憶 4,000 語である。プロセッサは 2600 語で普通の磁心記憶におかれる。文法上の制限は own array がないだけだといっている。テープ読取器は, コペンハーゲンと同じく Facit の 500 c/s のものを使っている。部屋のすみに 1,000 c/s のものがほりをかぶっておかれてあった。帰ってからわかったのであるが, 森口教授がここを訪問された折には Dijkstra が「これは自分が設計したもので, Facit の読取器より故障が少ない」と自慢したそうである。残念ながら Dijkstra が去るとともにお蔵入りとなったようだ。穿孔室では Book keeping とか, Music とかのプログラムを盛んに穿孔していた。

これら二つの計算センタは ALGOL の性格が大変似ている。そして一方には Naur がいて, 他方には Wijngaarden がいる。Naur は, はじめは Dijkstra 程に一般的な ALGOL を主張してはなかったようであるが, いまはだいぶ Dijkstra に傾倒しているようだ。Wijngaarden は, でっぷりふとった品のいいおじいちゃんで, はじめて会ったときは何だかどこかの社長さんみたいな感じであった。しかし討論をはじめると夢中になって, ときどき子供のようなかん高い声をだし仕事まで子供っぽくなってくる。こんな人たちが両方の計算センタの支配的人物であることからおしても, その性格がよくわかるようだ。要するに両者とも計算センタではなくて, プログラムの研究室なのである。そして互いに連絡を保ちながら, ALGOL 6Y の研究にのりだしている様子であって, さらにアムステルダムにいた Dijkstra や Wirth などが他の大学に去って, そこで同じことを追求しているのである。

WG 2.1 の会議のあとでは, マインツとダルムシュタットへ行った。マインツの Hochschule にはかつて Samelson と Bauer がいて, 逐次翻訳の理論をひろめ, ALGOL 屋たちの中心となっていたのである。丘の斜面に開かれた小都会の, ひろびろとした大学の構内に, キャップェに接続して建てられた小さな計算センタであった。Baur がマインツを去ってミュンヘンに移ったということは知っていたが, Samelson までミュンヘンに移ったことは手紙をもらうまでわからなかった。それで彼らの作ったプロセッサの前で, 彼ら自身の説明をきく計画は駄目になってしまった。私としては, 彼らの論文の中にかかれてある optimization を, どれだけやっているかに興味をも

っていたのであるが, 実際にマインツにはプロセッサの中味を知っている人は誰もいなかった。こんなことなら, Meeting の間に Samelson にすすめられたようにミュンヘン大学の計算センタへ行ってみればよかったのであるが, あとのまつりである。

マインツには例の ZUSE 22 というきわめて古くさい真空管の機械がある。このほかに Siemens 2002 があって, こちらの方は学生に使わせないらしい。ALGOL はどちらでも使うことができるが, GIER ALGOL 程一般的な文法はもっていない。もちろん procedure の recursive call はやることができない。助手の若い人にいろいろな話をきいて, 簡単なプログラムを作ってもらい Siemens 2002 ALGOL にとおしてもらった。この ALGOL は ALCOR グループの一環として開発されたものらしく, はじめに

'COMMENT' ALCOR MAINZ 2002, 5 K;

と印刷される。記憶装置はコア 12,000, ドラム 10,000 であるから 5 K の意味はわからない。日本では文字の delimiter を小文字, 大文字は letter と区別している例が多いが, ここでは delimiter symbol' を用いている。5 孔のテープで小文字はない。ところが GIER は 8 孔 (1 孔はパリティ) のテープを用い, 豊富な活字を使用しており, 文字の delimiter は本当に

begin

と穿孔するのである。しかもアンダラインは赤に印刷される。Naur は大変気軽にアンダラインと文字のキーを交互に叩いていたが, 日本のようなせちがらく, せかせかしたところでは, 考えただけでも反対されそうな方法である。

ダルムシュタットの Hoch Schule は Walther という大先生のいるところであり, でかける前に大変丁寧な手紙をいただいたものであるけれども, 行ってみたら, Dr. W. Barth という若いらしい人がただ一人で, あとは不在であった。もっともこの町の IBM 7090 と 1401 がある, すばらしく綺麗な計算センタを見学するように Dr. Walther がはからっておいてはくれたけれども, Barth はやはりコンパイラのことを何にも知らないが, よく使っているらしく文法のことにはくわしかつた。要するに own, recursive procedure はだめで, パラメータの complete specification, actual-formal の型の一致などを必要とする。I/O はあたかも宣言のような形に書いていく。さらに面白いのは label を宣言しなければいけないで, その

宣言はスイッチの形に作るのである。スイッチという宣言が、本当のスイッチとして使えるのかどうかは聞きそこねてしまった。Barth 氏のいる建物は、今にもくずれおちそうです暗く、じめじめした感じであった。計算機の所在をきくと、「ここにはない、別のところに Elliot 803 がある」という。あまり機械自体には興味はないらしい。フランクフルトへ帰る汽車の時間にだいふ間があるので、いろいろな話に脱線していたら「建物の中を案内してやる」という。あちらこちらほとんど無人の部屋々々をめぐっていくうちに、何と DERA という名の、本体で部屋の壁が作られている計算機にでくわした。ドアを開くと計算機のお腹の中に入るのである。Barth 氏はどうも DERA を計算機とはみとめていないらしい。女の人が一人机に向って腰かけていたから使っているらしいのであるが、何しろ古い計算機である。ZUSE とか DERA の要目を知りたい人は計算機の発達史をかなり丁寧に書いた書物を見てください。ここでは割愛する。

### 3. ALGOL の思想

マイッツはかつて ALGOL の中心地であったが、その文法にはかなりの制限項目がある。ダルムシュタットの ALGOL もそうであった。Fortran にくらべて簡単なコンパイラを必要としたマイッツやダルムシュタットのような条件の上で、ALGOL の思想が発生し、発達していったのであろう。これらのコンパイラにくらべて、コペンハーゲンやアムステルダムでの ALGOL の一般性のあざやかな相違に、ヨーロッパの ALGOL の短かいが充実した発達史が示されているようだ。

WG 2.1 の雰囲気からは ALGOL 研究の中心が Bauer や Samelson から, Wijngaarden, Dijkstra,

Naur などとうつったことがうかがわれる。ALGOL は当初から descriptive language としての性格をもってはいたが、Samelson, Bauer の論文からはむしろ Fortran 的な思想の方が強く感じられる。ALGOL 58 から 60 への発展は、一方において Wijngaarden, Dijkstra のグループによって、他方において Naur のグループによって強力に進められたらしい。そして今や、その行きつくところとして最も一般的な情報の記述言語の Kernel の追求がはじまっているのである。

訪問した計算センターも少なく聞いた話も非常に限られた側面のものであるから、上記の観察は全面的に主張できるわけではない。ヨーロッパの implementor たちが ALGOL を大変厳密に、高い次元で解釈しているようだ、という私の意見に、議長の Van der Poel が「私はそう思わない、WG 2.1 の中でも……」と答えたのは、しかしながら私の観察を支持するものであると、今は考えている。

大変「風船旅行」的な報告記になってしまい、読者にも、スポンサーにも、森口先生にも申訳ない次第であるが、あまりにも少ない日数で、私自身大変不満に思っているという弁介をさせて戴きたい。最後に ALGOL Bulletin の発行責任者が、このような仕事には全く無能力であつたらしい Dijkstra (彼は一回も Bulletin を出さなかった) から、Duncan の手にうつされたので、もし letter なり paper なりの形で何か書きたい方々のために Duncan の住所をお伝えする。

Mr. F.G. Duncan.  
Lubeckstraat 71,  
The Hague, Netherlands

(昭和 39 年 5 月 18 日受付)