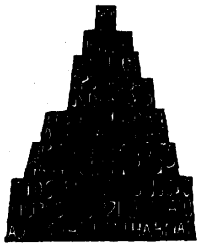

5. ALGOL 60/ALGOL 68†

川　合　　慧††

1. はじめに

「ALGOL 系言語」という言葉が示すとおり，ALGOL 60 はプログラム言語の発展の中で最も重要な役割を果たし，その後継言語である ALGOL 68 も言語の諸概念の研究に多大な貢献をしてきた．本解説では両言語の成立以降その改訂過程とその内容について概説する．

2. 成立とその特徴

ALGOL 60

今から約 20 年前，1958 年から 1962 年にかけて*設計[17)~19),21),22)]され，その後の 10 年間はほとんど唯一の 'きちんと' 定義されたプログラム言語として使用され，多くの後継言語の源流の役目を果たしてきた ALGOL 60 も，近来の言語研究の発展の渦の中で最近とみに影が薄くなってきたようである**．

ALGOL 60 の改訂報告書[19)]ではこの言語を '機械語への直接的な翻訳' が可能で '広範囲の数値計算を表現' できる '国際的なアルゴリズム言語' と規定している．

ALGOL 60 によって確立された言語上の概念には次のようなものがある．

宣言　プログラマが計算に使用する種々の対象（変数，配列，手続きなど）を陽に指定して導入する方式を定めた．これにより，誤りの少ない大規模プログラミングへの道が開けたほか，各種の量とそのためのデータ領域の実行時管理法が明確化された．

構造化文　複合文，条件文，反復文などの設定により構造的なプログラム作成が容易となった．

手続き　概念を明確化しパラメタ結合の方式の基本を定めた．特に名前呼び (call by name) の結合は構文要素の置換による強力なものであった．

今日ではごく当然のことと考えられているこれらの概念が，すべて ALGOL 60 によって確立されたものであることは注目に値することである．これらのしっかりとした概念は処理系の作成にも好影響を与え，特に欧州において盛んに使用される基盤となった．しかしまた，1960 年代の実行効率至上主義と，当時の最適化技術の水準の低さとから，機械語に強く密着した言語 FORTRAN に対して分が悪い面があったことも事実であった．入出力機能を言語に取り入れなかったことも，その一因となっている．

ALGOL 68

ALGOL 60 の後継言語として IFIP によって 1965 年から 1968 年にかけて設計された ALGOL 68 は，60 年代の種々の着想が盛り込まれた大規模な言語[27),29)]である．言語記述の手段として，ALGOL 60 での BNF に代わって文脈依存型の 2 段階文法[1)]が用いられたのも特徴のひとつである．ALGOL 68 の初版[27)]の主要な項目を以下に示す[12)]．

データ型　複雑な構造を持つデータ型を，配列，構造化，合併，参照，手続きという 5 種の構造化手段によって構成し，名前もつけられる方式を採用した．

参照値　ポインタを参照値として扱い，変数の概念をきちんと定式化した．

手続き　手続きもデータ構造化の一手段とした．パラメタ結合は値呼びに限られたが，参照値や手続き値を渡すことで名前呼びの効果もかなり実現される．

演算子　単項と 2 項の演算子を手続きの形式で定義可能とし，パラメタ数やその型が異なれば同一記号を複数の演算に対応させられるようにした．

並行処理　並行実行と同期のための構文を導入し

---

† ALGOL 60/ALGOL 68 by Satoru KAWAI (Department of Information Science, Faculty of Science, University of Tokyo).

†† 東京大学理学部情報科学科

* 当初の開発は特定の団体によるものではなく，ALGOL 開発を目的とした何回かの国際会議によって行われ，それを欧州の計算機学会や米国の ACM が支援する形式をとった．開発等の作業責任が IFIP の手に移ったのは，ALGOL 作業部会が 1962 年に作られてからである．

** 1958 年当初は数値計算用の言語が考えられた．文献 21) では 'algebraic language' とされている．Algorithmic Language (算法言語) からとられた ALGOL の名が登場するのは文献 22) からである．

た.

式言語 すべての構文要素が値を返すこととした.

自動型変換 文脈によって要求される型を得るための自明な型変換（整数→実数など）を行う.

標準前置部 入出力や標準手続きを正確に記述する部分（ALGOL 68 で記述）が追加された.

以上の諸項目からもわかるとおり，ALGOL 68 は ALGOL 60 に比べて飛躍的に強力な言語となっている．それだけに処理系にかかる負担も増大したわけであるが，欧州，特に設計者グループの周辺においては早くから処理系の作成が進められた．現在では，主な研究機関であれば ALGOL 68 の使用が可能となっている.

3. 最近の動向

ALGOL 60

改訂報告書作成以後の作業は IFIP の作業グループ WG 2.1 に引き継がれ，1964 年には部分言語[24]と入出力[23]に関する報告書が作成された．しかしそれ以後の WG 2.1 の活動の主力は ALGOL 68 として結実することになる新言語へ向けられた．これとは別に，ECMA*が 1965 年に部分言語[2]について，ACM が 1964 年に入出力[9]について，それぞれ別案を提示している.

1972 年には，改訂報告書を一部変更したものと部分言語および入出力の提案とをまとめたものが，ISO** によって推薦規格案[5]として作成されたが，IFIP はこれを正当なものとは認めなかった．それは，新しい概念や構文の持ち込みよりも言語の安定性のほうが重要視されたためである.

IFIP 自体も ALGOL 60 の '保守' に全く無関心だったわけではなく，C. A. R. Hoare を長とする小委員会が改訂報告書の説明の不備や解釈不明な問題点に検討を加えてきた．修整の原案は 1974 年に作られ最終的な修整報告書[15]が 1976 年に発表されている[14),16]．以下にその概要を示す[30].

1) 改訂報告書において未解決とされていた諸問題

関数の副作用 有効性のほうが害よりも重要視され，禁止はされなかった．また，構成要素の評価順序に値が依存する式（関数の副作用による）を含むプログラムは未定義とすることが明示された.

パラメタの型変換 算術型（整数と実数）のパラメタ結合に際しては，どちらの方向にも必要なら型変換が起こるものと規定された.

占有変数 すべての占有変数はプログラム全体を囲む仮想的なブロック（環境ブロック）において宣言され，0 または **false** に初期化される．したがって占有配列の上下限は整定数に限られ，動的な占有配列は禁止された．ただし，名前の有効範囲はプログラマが指定した本来のブロックに限られる.

反復文 step-until 要素を持つ反復文の定義が以下のように与えられ，刻み幅(B)と終端値(C)の評価時期と回数とが確定された[16].

for V:=A **step** B **until** C **do** S
=**begin** ＜Bの型＞θ; V:=A; θ:=B;
Γ: **if**$(V-C)\times sign(\theta)\geq 0$ **then**
begin S; θ:=B;
V:=$V+\theta$; **go to** Γ **end**
end

制御変数Vは単純変数に限られ，反復終了時の値も上の定義によって決められることになった.

規制と宣言との対応 手続きの規制部(specification)で与えられる仮パラメタの種類と実パラメタの形式との対応表が新設され，すべての仮パラメタの型と種類とを規制部に書くことが義務づけられた.

2) その他の修整点

整数名札 整数名札 (integer label) は廃止.

環境ブロック 以下に示す入出力手続きが環境ブロックにおいて宣言されていることになった．これらは整数パラメタで指定されるチャネルに対して動作する.

inchar, *outchar*, *outstring*, *outterminator*, *ininteger*, *outinteger*, *inreal*, *outreal*

このほかに以下に示す標準関数が含まれている.

abs, *iabs*, *sign*, *entier*, *sqrt*, *sin*, *cos*, *arctan*, *ln*, *exp*, *length*, *maxreal*, *minreal*, *maxint*, *epsilon*, *stop*, *fault*

部分言語 以下の 2 水準が定義された.

水準 1 占有変数なし．名前呼びの型変換なし．英字は大文字か小文字かのどちらかのみ．名前は先頭の 12 文字のみ有効.

水準 2 （水準 1 の制限に加えて）再帰呼出し禁止．名前呼びの実パラメタは名前 1 個あるいは文字列に限る．スイッチの行先並びは名札のみ．スイッチ，手続き，関数の各仮パラメタなし．多重代入文なし．
名前は先頭の 6 文字のみ有効.

前述の ISO による推薦規格案[5]は，結局正式の規格

* European Computer Manufacturer's Association
** International Organization for Standardization

となることなく1978年に取り下げられた．現在では上記の修整報告書を規格案とする作業が行われている．

日本においては ALGOL 60 を JIS とする作業が早くから始められ，1967年に第1版が完成した．部分言語については言語水準を5段階 (7000～3000) とすることとし，入出力の規格も対応して作られた．言語の内容としては，改訂報告書のそれに整数名札の禁止やパラメタ対応の表による記述などが加わったものであった．この規格は，修整報告書のもととなった議論などにより1972年に改訂された[6]．現在は次の改訂作業が進行中である．

ALGOL 68

ALGOL 68 については初版発表後間もなく改訂作業が始められた．ALGOL 60 の場合と同様に，言語自体の変更は避け，プログラムの書きやすさと意味記述の理解しやすさとが作業目標であった[12]．以下に改訂報告書 (1975年)[28] における主な改訂点を示す[7),25)]．

拡張表記の正式構文化 初版においては言語の厳密な意味記述に重点が置かれ，'プログラムの書きやすさ' は一種のパタンマッチマクロである拡張 (extension) 記法に頼っていた．たとえば初版における変数宣言は

ref real x = **loc real** := 3.14

という書き方が正式で，領域の確保 (**loc**) や名前がポインタであること (**ref**) および等価宣言 (=) という '意味' を忠実に表現しているのに対し，その拡張

real x := 3.14

は ALGOL 60 のそれに似た書きやすく読みやすいものとなっていた．反復構文や選択構文もこれと同様な扱いであった．改訂報告書では拡張のほとんどすべてが正式な構文となり，IF 文を含むすべての構造化文には宣言の有効範囲を局所化する機能も与えられた．

合併型の扱い 合併型 (union) は複数の型の値の和集合を値とする型である．初版では，この型の変数の '現在値の型' を知る方法として型比較演算子 (::) を用意した．たとえば **union** (**int, char**) 型の変数 v の値を '増加' させるのに

v := **if int** :: v **then** v+1
else repr (**abs** v+1) **fi**

などとする (**repr** と **abs** は整数と文字間の変換)．改訂版ではこの悪用されやすい演算子のかわりに，Hoare が提案していた型選択構文[4]を導入した．

v := **case** v **in**
(**int** i): i+1,
(**char** c): **repr**(**abs** c+1)
esac

記述法の改良 初版では，名前の出現とその宣言との対応や多重宣言の禁止などが文脈条件 (context condition) として平文で記述されていた．改訂版では，一次元表現の名前表を表す構文変数 **NEST** を新設し，文脈条件をすべて生成規則の範囲内で処理するようにした．宣言局所化の記述の問題も，**NEST** の中に壁の印 **new** を含めることで解決されている．また，ふつうは注意書きで処理される細かい規則や制限事項を生成規則で処理するために，述語 (predicate) と呼ばれる仕組みを導入した．これは **where** や **unless** で始まり **balance** や **follow, count** といった種々の性質語 (と目的語) を含む構文で，成立すれば空列に置換されるがそうでなければ生成規則が定義されない．すなわち，条件が不成立である述語を含む構文からは最終的なプログラムは生成されない．これを生成規則に含ませておけば目的は達せられる．

ALGOL 68 についてのその後の動きとしては，合併型や並行処理の機能を取り除いた部分言語[9),10)]の仕様決定と，分割処理方式の提案[11]とがある．後者は，木構造分割を行うトップダウン的なものと，前・後処理つきのライブラリを呼ぶ形式のボトムアップ的なものとを実現しようとするものであるが，まだ種々の議論を行っている段階である．言語の規格化の動きとしては，ALGOL 68 の開発を担当した IFIP の WG 2.1 が，ISO に規格化の手順を問い合わせているというのが現状である．処理系の例は欧米に多く，かなりの広まりを見せているが[13),20),26)]，日本では英国から移植された ALGOL 68C[7),8)]が東京大学と電気通信大学で稼動しているのみである．

4. おわりに

名前呼びという，ほかの言語に類を見ない強力な機能を有する ALGOL 60 も，実用面ではほかの後続言語に座を明け渡したといってよい．今後は言語研究の原点としての役割を強めてゆくであろう．これに対し ALGOL 68 はまだ壮年期であり，システム記述や大規模ソフトへの適用も多い．これからも実用的な重要言語として使用されてゆくことと思われる．

参考文献

1) Cleaveland, J. C. and Uzgalis, R. C.: Grammars for Programming Languages, Elsevier, North-Holland (1977).

2) ECMA Subset of ALGOL 60, CACM, Vol. 6, pp. 597-599 (1963).
3) Hibbard, P. G.: A Sublanguage of ALGOL 68, Sigplan Notices, Vol. 12, No. 5, pp. 71-79 (1977).
4) Hoare, C. A. R.: Notes on Data Structuring, in Structured Programming, Academic Press, London (1972).
5) ISO Recommendation R 1538, Programming Language ALGOL (1972).
6) 日本工業規格, 電子計算機プログラム用言語 ALGOL (水準 7000, 6000, 5000, 4000, 3000) および入出力 (水準 70, 60, 50, 40, 30) (1972).
7) 川合 慧: ALGOL 68 とその処理系, 情報処理, Vol. 21, pp. 1162-1173, 1266-1278 (1980).
8) Kawai, S. and Ishihata, K.: A Transportation of Multiphase Compiler, J. of Information Processing, Vol. 2, pp. 143—145 (1979).
9) Knuth, D. E.: A Proposal for Input-Output Conventions in ALGOL 60, CACM, Vol. 7, pp. 273-283 (1964).
10) Lindsey, C. H.: Some ALGOL 68 sublanguages, in ALGOL 68 Implementation (J. E. L. Peck ed.), North-Holland (1971).
11) Lindsey, C. H. and Boom, H. J.: A Modules and Separate Compilation facility for ALGOL 68, Algol Bulletin No. 43, pp. 19-53 (1978).
12) Lindsey, C. H.: and van der Meulen, S. G.: Informal Introduction to ALGOL 68, North-Holland (1971) and revised edition (1977).
13) McGettrick, A. D.: Algol 68 a first and second course, Cambridge University Press (1978).
14) DeMorgan, R. M., Hill, I. D. and Wichmann B. A.: A Supplement to the ALGOL 60 Revised Report, Computer J., Vol. 19, pp. 276-288 (1976).
15) DeMorgan, R. M., Hill, I. D. and Wichmann' B. A.: Modified Report on the Algorithmic Language ALGOL 60, Computer J., Vol. 19, pp. 364-379 (1976).
16) DeMorgan, R. M., Hill, I. D. and Wichmann, B. A.: Modified ALGOL 60 and the step-until element, Computer J., Vol. 21, p. 282 (1978).
17) 中島勝也: ALGOL 60 の改訂について, 情報処理, Vol. 4, pp. 149-153 (1963).
18) Naur, P. (ed.): Report on the Algorithmic Language ALGOL 60, CACM, Vol. 3, pp. 299-314 (1960). 邦訳: 淵・伊藤訳: 算法言語 ALGOL 60 に関する報告, 情報処理, Vol. 1, pp. 157-162, 219-227 (1960), Vol. 2, pp. 30-38 (1961).
19) Naur, P. (ed.): Revised Report on the Algorithmic Language ALGOL 60, CACM, Vol. 6. pp. 1-17 (1963).
20) Pagan, F. G.: A Practical Guide to Algol 68, John Wiley (1976).
21) Perlis, A. J. and Samelson, K. (eds.): Preliminary report-International Algebraic Language, CACM, Vol. 1, p. 8 (1958).
22) Perlis, A. J. and Samelson, K. (eds.): Report on the Algorithmic Language ALGOL by the ACM Committee on Programming Language and the GAMM Committee on Programming, Numerische Mathematik, Vol. 1, pp. 41-60 (1959).
23) Report on Input-Output Procedures for ALGOL 60, CACM, Vol. 7, pp. 628-629 (1964).
24) Report on Subset ALGOL 60, CACM, Vol. 7, pp. 626-627 (1964).
25) Tanenbaum, A. S.: A Comparison of PASCAL and ALGOL 68, Computer J., Vol. 21, pp. 316-323 (1978). 邦訳: Pascal と Algol 68 の比較 (上, 下), bit, Vol. 12, No. 1, pp. 24-33, No. 2, pp. 22-31 (1980).
26) Tanenbaum, A. S.: A Tutorial on ALGOL'68, Computiug Surveys, Vol. 8, pp. 155-190 (1976).
27) van Wijngaarden, A., et al. (eds.): Report on the algorithmic language ALGOL 68, Numerische Mathematik, Vol. 14, pp. 79-218 (1969).
28) van Wijngaarden, A., et al. (eds.): Revised report on the algorithmic language Algol 68, Acta Informatica, Vol. 5, pp. 1-236 (1975) and Springer-Verlag, Berlin (1976).
29) 米田信夫: 新算法言語 ALGOL 68, 数理科学, Vol. 7, No. 6 (1969)-Vol. 8, No. 7 (1970), ダイヤモンド社.
30) 米田信夫, 野下浩平: ALGOL 60 講義, 共立出版 (1979).

(昭和 56 年 2 月 6 日受付)