



8. SNOBOL†

角 田 博 保††

1. はじめに

SNOBOL 族の言語には開発順に, SNOBOL¹⁾, SNOBOL 2, SNOBOL3²⁾, および SNOBOL4³⁾ がある. 本文では現在の主流である SNOBOL4 を中心に解説する. 歴史的な部分についてはおもに文献⁴⁾ を参考にした.

原 SNOBOL (StriNg Oriented symBolic Language) は 1962 年ベル電話研究所 (以下 BTL と略) の Farber, Griswold, および Polonsky によって開発された. SCL (The Symbolic Communication Language, 非公開) と COMIT⁵⁾ を先祖とし, 文字列で表現されるのが自然であるような非数値的科学的データ (たとえば数式) を取り扱うための汎用言語として, 計算機を研究用に利用しようとして考案された. 最初の処理系は 1963 年に IBM 7090 上に作成され, テキスト生成, グラフ解析, 構文解析等に使われた. その後, 組み込み関数機構を追加することで SNOBOL 2 (1964 年) が, さらに関数定義機構を組み込むことで SNOBOL 3 (1964 年) が開発された. この SNOBOL 3 は SNOBOL 2 から始まった改訂作業のある意味での最終版に当たるもので, IBM 7090/94 上に処理系が作られ, BTL の内外でよく使われた.

1965 年頃始まった第 3 世代計算機の導入にともなうソフトウェアの移行 (移植) にあわせて, SNOBOL 3 を大幅に改訂した SNOBOL 4 の設計が始まった. 当初から移植性のある処理系作りがおこなわれ 1966 年に最初の処理系が完成した. その後, 処理系の利用経験を生かして言語機能の追加, 削除がおこなわれ, 1969 年に最終版が完成した. 1971 年 Griswold がアリゾナ大学へ移った後は SNOBOL 4 の開発作業は終り, それ以降は新機種用処理系開発とより高い効率を得るための技法開発がおもにおこなわれている.

2. 最近の動向

● 言 語

SNOBOL 族の言語設計方針は専門プログラマでない人でも簡単にプログラムできるように, その使いやすさに重点が置かれ, ①簡素さ (語いが少ないこと), ②簡単さ (学習, 理解が容易であること), ③問題向き, および④融通性を目標にしている点で一貫している. ここでは SNOBOL 4 言語の特徴をそれまでの SNOBOL 族との係わりのもとに説明する.

原 SNOBOL の特徴である, ①使用者が領域管理を気にせずに自由に可変長の文字列をデータとして (宣言なしで) 扱えること, ②文字列処理にパターン照合を使うこと, ③間接参照演算によって実行時に変数名を生成できること, および④構文が単純であることは SNOBOL 4 においても同様にあてはまる. SNOBOL 族のプログラムは文の列から構成されている. 各文の実行結果には成功と失敗の別があり, 条件付き実行指定によって次に実行する文を選択できるようになっている. 一般に使われる制御構造や予約語はない.

SNOBOL 4 においてそれまでと大幅に内容が変わっているのはパターン照合である. SNOBOL 3 まではパターンは構文的に区別される数種のパターン要素の列としてあらわされた (マルコフのアルゴリズム⁶⁾ の規則を表現するには十分である). SNOBOL 4 ではパターンはパターン型のデータとして扱われる. 10 数種の基本パターンといくつかの合成操作によって複雑なパターンを (実行時に) 構成することができる. また非評価式によってパターン照合時に動的にパターンを構成することも可能である (これによって再帰的なパターンも構成できる).

パターン型とともにいくつかの型が導入されている (SNOBOL 3 までは文字列型しかなかった). 中でも特徴的なのは配列, 表 (文字列による索引機構), およびプログラム定義データ (レコードに対応) である. これらによってデータ構造の表現力が大幅に向上している.

† SNOBOL by Hiroyasu KAKUDA (Department of Information Science, Tokyo Institute of Technology).

†† 東京工業大学理学部情報科学科

使用者が自由に関数を定義できる機構は SNOBOL3 において導入された。この定義関数機構の特徴は、簡単さと融通性のために宣言方式ではなく実行時定義方式になっている点である。関数名、引数名、局所の変数名、および定義本体の開始ラベル名の指定によって動的に関数を構成するものである。この導入のしかたは一般のプログラミング言語の志向している方向と矛盾している。すなわち、定義範囲の宣言やラベルの局所性がないことから、構造的なプログラムや分割コンパイルモジュールがほとんど作れなくなっている。

連想方式による入出力機構も特徴的である。これは変数名に入出力連想を付け、入力連想の場合は変数の評価時に、出力連想の場合は変数への代入時にそれぞれ入力、出力が起こるというものである（暗黙の処理と呼ばれる）。

実行時コンパイル機能は SNOBOL4 において導入された機能である。これによって融通性が飛躍的に向上した。

SNOBOL4 のプログラム例については文献^{7)~9)}を参照されたい。

● おもな処理系

SNOBOL4 には SIL (SNOBOL4 Implementation Language) という中間言語（マクロ言語）によって記述された処理系が Griswold らによって開発されている¹⁰⁾。これは SNOBOL4 の事実上の規格とみることができる。SIL を対象機種種の機械語にマクロ展開することで、簡単に SIL 版 SNOBOL4 が作成可能である（現時点で 40 以上の機種で実働化されている）。SIL 版 SNOBOL4 は一般に処理速度が遅く、所要記憶容量が多い。

SIL 版とは別に高速性を追究して作られた処理系として有名なものに、SPITBOL¹¹⁾ と SITBOL¹²⁾ がある。SPITBOL は最初の実用コンパイラで SNOBOL4 の機能のほとんどと若干の拡張機能を備えており、IBM 360 上に作られた。IBM 360 用 SIL 版と比べて 3~10 倍の処理速度が得られている。現在は MACRO SPITBOL¹³⁾ という高速性を維持したもとで移植可能にした処理系が開発されている。SITBOL は実用として使える解釈型処理系として PDP-10 上に作られた。SNOBOL4 と SPITBOL の両方の機能を含んでいる。これら SIL によらない処理系は方言も含めて 50 以上開発されている。SNOBOL4 の機能を拡張したものとしては SNOBOL4B¹⁴⁾（3 次元文字を扱う）が有名である。

● 具体化手法

SNOBOL4 具体化における問題点は、①領域管理、②実行時の名前表管理、③連想機構、④パターン照合、⑤実行時コンパイル機構、および⑥遅延束縛（非評価式および実行時の関数名変更等）である。文献¹⁰⁾では SIL の具体化手法が全般にわたって詳しく解説されている。領域管理については種々の手法が開発された^{10), 13), 15)}。MACRO SPITBOL では間接解釈コードを使って連想機構、遅延束縛を効率的に処理している。

● 適用分野と使用の現状

原 SNOBOL が SNOBOL4 へと改訂されるにしたがって、文字列処理から広範囲の非数値処理へと適用分野が拡大された。現在はドキュメントの清書化に使われたり、古典文学の分野で利用されているという報告がある⁴⁾。当初志向していた数式処理の分野では実用的には使われていない。また、コンパイラ作成用に実験的に使われることはあるが、実用性は疑わしい。SNOBOL4 は機能が大きくプログラミングがしやすいことから、実験的な利用または道具としてよく利用されている。SNOBOL4 利用者に対するアンケート¹⁶⁾の中で SNOBOL4 に興味がある点として、応用（173 件）、教育（133 件）、実働化（112 件）、言語設計用（101 件）という結果が得られている。また使われている処理系は SIL 版（108 件）、SPITBOL（83 件）、MACRO SPITBOL（49 件）、SITBOL（18 件）等であった。

日本国内においてはいくつかの大学で SNOBOL3 が使われている^{17), 18)}。ファイル転送といった簡単な道具からプリプロセッサのようなやや重い道具まで実用的に利用されている。また、日本語言語処理の教育に利用されている例もある¹⁹⁾。SNOBOL4 においても SIL 版の処理系が作成されたり²⁰⁾、ミニコン用に処理系が開発されたりしている^{21), 22)}。SIL 版もいくつかの計算機環境のよい施設で利用されている。大学の情報関連専門学科に対するアンケート調査²³⁾では、SNOBOL 族の整備度は 30%、利用度は 7% であり、現在はないが必要としている割合は 30% と報告されている。

● 実用性の評価

SNOBOL4 は記述能力が大きい。使用者は領域管理とか戻り処理(backtracking)とかを考えずにパターンという形で照合したいことを簡単に記述することができる。このパターンの記述能力は Turing 機械ま

でパターンによって表現可能なほどである。表型の導入は名前表のような構造が簡単に利用できるようになった点で効果大きい。また、実行時コンパイル機構は対話型処理をするのに役立っている。逆に欠点としては、パターン照合が左から右のみであり、照合位置の自由な移動がおこなえない点、統合機能がない点、パターンの語が多過ぎ、特殊で一般性がない点、基本パターンを増す機能がない点等があげられる。SNOBOL 4 の構文は構造的プログラミング派と相容れないが、素人利用者でも簡単に使えるということを見ればこの構文の適した利用分野もあるのだと思われる。

効率の面からみれば SNOBOL 族は高速化できない特質をいくつも持っている。SIL 版は大きく遅いが、今日の記憶容量が多く演算速度の速い計算機環境では、簡単な道具として使うには十分役立つ。SIL 版より 1 桁ほど高速であるといわれている SPITBOL がある環境では、より複雑な仕事も実用的におこなうことができる。

● 他言語との関係と比較評価

SNOBOL 流のパターン照合を利用したマクロ処理系(たとえば Stage 2, ML/I)はいくつか作られたが、プログラム言語の主流に対してはあまり影響を及ぼしていない(動的な文字列を扱える言語は非常に少ない)。SNOBOL 4 の後継をねらって作られた言語として SL 5²⁴⁾ と Icon²⁵⁾ があげられる。SL 5 は SNOBOL 4 に制御構造を入れ、手続き呼び出し機構を一般化したもので、研究的目的で開発された。Icon は逆戻り処理用基本操作(generator)を導入することで SNOBOL 4 のパターン照合処理に対応する機能を Icon 上で記述できるようにした点が注目される。SNOBOL 4 はパターン照合言語と基底言語の 2 言語から構成されているといえるが、Icon はそれを 1 つの言語に統一したとみることができる²⁶⁾。その点で Icon の方が進歩していると考えられるが、SNOBOL 4 がパターンという形で何を照合したいかを記述するのに対し、Icon ではどう照合するかという手続きを記述することになり、パターンの持っていた使いやすさを越えることができるかどうかが問題である。

● 理論的側面

文献²⁷⁾では SNOBOL 4 の意味記述を Scott 流におこなっている。パターン概念の形式的取り扱いが文献²⁸⁾でおこなわれた。

● 現実的情報

SNOBOL 4 Information Bulletin はアリゾナ大学の Griswold 教授によって発行されている(宛先: Professor Ralph E. Griswold, Department of Computer Science, The University of Arizona, Tucson, Arizona 85721, U.S.A.). SNOBOL 4 のドキュメントも同所から入手可能である。また、IBM 360/370 用 SIL 版 SNOBOL 4 も磁気テープ(9トラック, 800 BPI または 1600 BPI, 800 BPI なら 2400 ft., 1600 BPI なら 1200 ft.)を送れば折り返し供給される(以上いずれも 1976 年 7 月 28 日付 SNOBOL 4 Information Bulletin では無料と表示されている)。

SPITBOL に関する情報は Dewar Information Systems Corporation (221 West Lake Street, Oak Park, Illinois 60302, U.S.A.) で一括して扱われている。こちらは有料のソフトウェアである。

3. 今後の見通し

SNOBOL 4 が現在の形に定まってから 10 年以上になるが仕様の変更もなく利用し続けられている。新機種が登場すれば対応した処理系が作られるという状況である(最新は VAX-11/780 と Prime-650 用 SIL 版の処理系)。SNOBOL 4 はそれが役に立つ分野で信奉者を作りやすい傾向があるといわれている²⁹⁾。その点でもまだまだ利用されることであろう。文献⁴⁾では、どんどん利用層の拡大していく小さな計算機上での処理系の成否が SNOBOL 4 の存続を決定するであろうと述べている。

謝辞 本稿をまとめるにあたり木村泉助教授から数数の助言を頂いた。ここに深く感謝します。

参考文献

- 1) Farber, D. J., Griswold, R. E. and Polonsky, I. P.: SNOBOL, a string manipulation language, J. ACM, Vol. 11, No. 1, pp. 21-30 (Jan. 1964).
- 2) Farber, D. J., Griswold, R. E. and Polonsky, I. P.: The SNOBOL 3 Programming Language, Bell System Technical Journal, Vol. 45, pp. 895-944 (July-Aug. 1966).
- 3) Griswold, R. E., Poage, J. F. and Polonsky, I. P.: The SNOBOL 4 Programming Language, Second Edition, 256 p., Prentice-Hall, Englewood Cliffs, N. J. (1971).
- 4) Griswold, R. E.: A History of the SNOBOL Programming Languages, ACM SIGPLAN Notices, Vol. 13, No. 8, pp. 275-308 (Aug. 1978).
- 5) M. I. T. Press: An Introduction to COMMIT

- Programming, Cambridge, MA. (1962).
- 6) Galler, B. A. and Perlis, A. J.: A View of Programming Languages, 282 p., Addison-Wesley, Reading, Mass. (1970) の Chapter Algorithms pp. 1-93.
 - 7) Griswold, R. E. and Griswold, M. T.: A SNOBOL 4 Primer, Prentice-Hall, Englewood Cliffs, N. J. (1973).
 - 8) Griswold, R. E.: String and List Processing in SNOBOL 4, 287 p., Prentice-Hall, Englewood Cliffs, N. J. (1975).
 - 9) Gimpel, J. F.: Algorithms in SNOBOL 4, 487 p., John Wiley and Sons, New York, NY. (1976).
 - 10) Griswold, R. E.: The Macro Implementation of SNOBOL 4, 310 p., Freeman, W. H. San Francisco, Calif. (1972).
 - 11) Dewar, R. B. K.: SPITBOL Version 2.0, (SNOBOL 4 Project Document S4D23), Illinois Institute of Technology, Chicago, IL. (Feb. 12, 1971).
 - 12) Gimpel, J. F.: SITBOL Version 3.0, SNOBOL 4 Project Document S4D30 b, Bell Laboratories, Holmdel, NJ. (June 1, 1973).
 - 13) Dewar, R. B. K. and McCann, A. P.: Macro SPITBOL—a SNOBOL 4 Compiler, Software-Practice and Experience, Vol. 7, pp. 95-113 (1977).
 - 14) Gimpel, J. F.: Blocks—A New Datatype for SNOBOL 4, Comm. ACM, Vol. 15, No. 6, pp. 438-447 (1972).
 - 15) Hanson, D. R.: Storage Management for an Implementation of SNOBOL 4, Software-Practice and Experience, Vol. 7, pp. 179-192 (1977).
 - 16) Griswold, R. E.: SNOBOL 4 Information Bulletin, S4B23, The University of Arizona, Tucson, Arizona (June 9, 1980).
 - 17) 角田博保: べたづめ方式の SNOBOL 3 処理系における性能測定, 第 16 回プログラミングシンポジウム報告集, 情報処理学会, pp. 197-202 (1975).
 - 18) 角田博保: べたづめ方式 SNOBOL 3 処理系の移植について, 情報処理学会記号処理研究会資料 4-2, pp. 1-10 (1978 年 5 月).
 - 19) 水谷静夫: SNOBOL と言語処理, 東京女子大学, 第 2 版, 61p. (1976 年 9 月).
 - 20) 白濱律雄: SNOBOL 4 処理系の移植経験, 情報処理学会計算言語学研究会資料 9-3 (1977 年 3 月).
 - 21) 大村博道, 浦 昭二, 大駒誠一: ミニコンのための SNOBOL 4 プロセッサの開発, 情報処理学会 第 17 回全国大会講演論文集, pp. 437-438 (1976).
 - 22) 中西正和, 小川原倫子, 植田健治: LISP による SNOBOL 4 の試作, 情報処理学会 第 20 回全国大会講演論文集, pp. 175-176 (1979).
 - 23) 田中幸吉, 工藤英男: 理工系情報学科における教育用電子計算機の現状と問題点, 情報処理, Vol. 21, No. 3, pp. 246-254 (1980 年 3 月).
 - 24) Griswold, R. E. and Hanson, D. R.: An Overview of SL5, SIGPLAN Notices, Vol. 12, No. 4, pp. 40-50 (1977).
 - 25) Griswold, R. E., Hanson, D. R. and Korb, J. T.: The Icon Programming Language: An Overview, SIGPLAN Notices, Vol. 14, No. 4, pp. 18-31 (1979).
 - 26) Griswold, R. E. and Hanson, D. R.: An Alternative to the Use of Patterns in String Processing, ACM Trans. Program. Lang. Syst., Vol. 2, No. 2, pp. 153-172 (1980).
 - 27) Tennent, R. D.: Mathematical Semantics of SNOBOL 4, In Conference Record of ACM Symposium on Principles of Programming Languages, pp. 95-107, Boston, MA. (1973).
 - 28) Gimpel, J. F.: A Theory of Discrete Patterns and their Implementation in SNOBOL 4, Comm. ACM, Vol. 16, No. 2, pp. 91-100 (1973).

(昭和 56 年 2 月 13 日受付)