by a translation subroutine. A new subroutine that effects certain scanning and related operations and which can be called in FORTRAN programs has been coded recently [15]. This SAPST subroutine adopts the syntax machine approach of Glennie [16] and can be used to deal with expanded meta-vocabularies. Scanning time, particularly of verbal material, depends on comparisons of quotations in the definition table with portions of the input string. Simultaneous comparisons of input material with several quotations would speed some scans considerably—a computer with several accumulators would be suited to this type of work.

Publication of this material has been delayed for some years in the hope that the implications of scanning and syntax definition would become clearer. There is no doubt now that syntax description and analysis are important, and they may provide new bridges between computing and biology [6] and between science and scholarship.

ACKNOWLEDGMENTS

The author would like to thank M. J. Bailey, G. Barth, E. J. D. Carter, G. F. Coulouris, Mrs. J. P. Jodeit, K. L. Kelley, Mrs. J. J. Levine, and S. C. Plumb for their assistance in various aspects of this work, which forms part of a program of research that is supported by the National Science Foundation, the Army, the Navy and the Air Force, and the International Business Machines Corporation.

REFERENCES

1. BARNETT, M. P. SSMTG programming notes nos. 26 and 27; see also BARNETT, M. P., AND FUTRELLE, R. P. MIT SSMTG quarterly progress report no. 33, 44 (1959); CARTER, E. J. D., AND FUTRELLE, R. P. SSMTG programming note no. 39.
2. BACKUS, J. The syntax and semantics of the proposed international algebraic language of the Zurich ACM-GAMM conference. Proc. First International Conf. Information Processing, UNESCO, Paris (1960).
3. IRONS, E. T. A syntax directed compiler for ALGOL-60. *Comm. ACM*, *4* (1961), 51.
4. BROOKER, R. A. AND MORRIS, D. J. A general translation program for phrase structure languages. *J. ACM 9* (1962), 1.
5. BARNETT, M. P. Low level language subroutines for use within FORTRAN. *Comm. ACM 4* (1961), 492.
6. BARNETT, M. P. Comments suggested by a consideration of computers. In Schmitt, F. O. (Ed.) *Macromolecular Specificity and Biological Memory*, 24 (MIT Press, 1961).
7. FARBER, D. J. Private communication.
8. PLUMB, S. C. Application of macro-programming techniques to circuit analysis with digital computers. M. S. Thesis, Univ. Vermont (1960).
9. BARNETT, M. P. AND KELLEY, K. L. Computer editing of verbal texts. Part I. The ES1 system. *Amer. Documentation*, in press. C. C. L. technical note No. 2 (1961).
10. CARTER, E. J. D.; COULOURIS, G. F. AND FUTRELLE, R. P. SSMTG programming note no. 25.
11. BAILEY, M. J.; BARNETT, M. P. AND CARTER, E. J. D. SSMGT programming note no. 34.
12. BAILEY, M. J. AND CARTER, E. J. D. SSMTG programming note no. 40.
13. BAILEY, M. J. SSMTG programming note no. 46.
14. BARTH, G. SSMTG programming notes nos. 43 and 49.
15. BAILEY, M. J. Unpublished work.
16. GLENNIE, A. E. Computation Centre report. Carnegie Institute of Technology (1960).

On Ambiguity in Phrase Structure Languages

Robert W. Floyd
Computer Associates, Inc., Woburn, Massachusetts

Let a *phrase structure language* be defined informally as a language defined by a set of definitions in the Backus notation [1] (more formal definitions may be found in [2, 3, 4]). The set of definitions is itself called a *phrase structure grammar*. A phrase structure grammar is *ambiguous* if there exists a sentence in the defined language which may be assigned more than one structure consistent with the grammar. Such an ambiguity in ALGOL 60, for example, is

$$\textbf{if } \beta_1 \textbf{ then } \phi \textbf{ if } \beta_2 \textbf{ then } \Sigma_1 \textbf{ else } \Sigma_2$$

where β represents any Boolean expression, ϕ a **for** clause, and Σ an unconditional statement.

Attempts have been made to devise an algorithm to determine for an arbitrarily given phrase structure grammar whether or not it is ambiguous. Such attempts may yield useful heuristic procedures, but, as will be shown below, a general algorithm does not exist.

Consider two sets of strings $\{x_i\}$ and $\{y_i\}$ over an alphabet, where i ranges from 1 to k. Let n_i signify the digit string which represents the integer i in decimal notation. Define a language by the following grammar:[1]

$$S_1 \sim x_1 \,;\, n_1 \cup x_2 \,;\, n_2 \cup \cdots \cup x_k \,;\, n_k \cup x_1S_1 \,,\, n_1 \cup x_2S_1 \,,\, n_2 \cup \cdots \cup x_kS_1 \,,\, n_k$$

$$S_2 \sim y_1 \,;\, n_1 \cup y_2 \,;\, n_2 \cup \cdots \cup y_k \,;\, n_k \cup y_1S_2 \,,\, n_1 \cup y_2S_2 \,,\, n_2 \cup \cdots \cup y_kS_2 \,,\, n_k$$

$$\text{sentence} \sim S_1 \cup S_2$$

Then S_1 contains those strings of the form

$$x_{i_1}x_{i_2}x_{i_3} \cdots x_{i_j} \,;\; n_{i_j} \,,\, n_{i_{j-1}} \,,\, \cdots \,,\, n_{i_2} \,,\, n_{i_1}$$

with $j \geqq 1$, and S_2 contains those strings of the form

$$y_{i_1}y_{i_2}y_{i_3} \cdots y_{i_j} \,;\; n_{i_j} \,,\, n_{i_{j-1}} \,,\, \cdots \,,\, n_{i_2} \,,\, n_{i_1}$$

with $j \geqq 1$. Each string in S_1 (or S_2) has only one syntactic structure consistent with the definition. Now it is readily seen that the language is ambiguous if and only if there is a string belonging to both S_1 and S_2, i.e., if there is a non-empty sequence of integers $i_1, i_2, \cdots, i_j$ such that $x_{i_1}x_{i_2} \cdots x_{i_j}$ is the same string as $y_{i_1}y_{i_2} \cdots y_{i_j}$. For an arbitrary choice of the paired sets $\{x_i\}$ and $\{y_i\}$, however, it has been proven that no decision procedure (algorithm) exists to determine in each case whether such a string exists [3]. Thus there can be no algorithm to determine in each case the ambiguity of a phrase structure grammar.

Certain grammars may be proven unambiguous. Whatever axiom system and rules of operation may be allowed for such proofs, the proofs themselves may be enumerated by an appropriate algorithm. The distinct ways of generating strings in a phrase structure language may also be enumerated. Consider an algorithm which alternates between proving languages unambiguous and generating the strings of the given language. If the given language

(Please turn to page 534)

[1] The symbol $\cup$ is used for union of two classes of strings, rather than the symbol | used in ALGOL 60 for reasons peculiar tothat langugae.

derived from history, items related to physical and laboratory examinations are also included. On the pages of the Information Code, the code numbers appear alongside the data to which they are assigned. Thus, the same pages can be used to record and also to transfer the data to punched cards and tape, making the data available for analyses including various statistical and significance measurements. The patient's Past Illness examination and Family History codes contain spaces for entering Standard Nomenclature Code numbers of all diseases. If the specific disease is not known, the general area of illness can be recorded and coded (e.g., "The father has arteriosclerosis" would be coded as 460-942). Other sections contain blank code spaces for insertion of additional data which may be of interest to individual investigators.

The fact that the Information Code is quite large would preclude its use in recording medical data routinely. However, it could be used to record data in special studies, either from hospital records or from patient interviews. In addition, individual sections of the code could be removed and used separately to record data in areas of special interest without using the entire code. Additions and modifications of the input data can also be made without changing the code numbers. In its present form, the Information Code would not enable patients to record information directly on the code sheets, and it would best be used by physicians.

ACKNOWLEDGMENT

We thank Carl Berkley and Dr. V. K. Zworykin for their aid in initiating this project, and Mrs. C. Bogner and Mrs. P. Smith for assistance in recording and compilation of the data.

		CARD #48
P. BLOOD CHEMISTRIES		
Amylase (units)		
Spec. I	□ □ □	Col. 57–59
" II	□ □ □	Col. 60–62
" III	□ □ □	Col. 63–65
" IV	□ □ □	Col. 66–68
Bilirubin Total (mg %) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 69–71
" II	□ □ □	Col. 72–74
" III	□ □ □	Col. 75–77
" IV	□ □ □	Col. 78–80
		CARD #49
History no.-Card no.		Col. 1–10
Bilirubin, direct (mg %) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 11–13
" II	□ □ □	Col. 14–16
" III	□ □ □	Col. 17–19
" IV	□ □ □	Col. 20–22
Bilirubin, indirect (mg %) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 23–25
" II	□ □ □	Col. 26–28
" III	□ □ □	Col. 29–31
" IV	□ □ □	Col. 32–34
Bromsulphalein (% ret.)		
Spec. I	□ □	Col. 35–36
" II	□ □	Col. 37–38
" III	□ □	Col. 39–40
" IV	□ □	Col. 41–42
		CARD #50
Phosphatase, Alkaline (units) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 71–73
" II	□ □ □	Col. 74–76
" III	□ □ □	Col. 77–79
Blank		Col. 80
		CARD #51
History no.-Card no.		Col. 1–10
Phosphatase, Alkaline (units)		
Spec. IV	□ □ □	Col. 11–13
Protein, total (gms %) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 14–16
" II	□ □ □	Col. 17–19
" III	□ □ □	Col. 20–22
" IV	□ □ □	Col. 23–25
Albumin (gms %) (3 dig. 1 dec.)		
Spec. I	□ □ □	Col. 26–28
" II	□ □ □	Col. 29–31
" III	□ □ □	Col. 32–34
" IV	□ □ □	Col. 35–37

FIG. 7

REFERENCES

1. LIPKIN, M., ENGLE, R., DAVIS, B., ZWORYKIN, V., EBALD, R., SENDROW, M., AND BERKLEY, C. Digital computer as aid to differential diagnosis. *Arch. Internal Med.* **108** (1961), 56.
2. WARNER, H. R., TORONTO, A. F., VEASEY, L. G., AND STEPHENSON, R. A mathematical approach to medical diagnosis. Application to congenital heart disease. *J. Amer. Med. Assoc.* **177** (July 22, 1961), 177–83
3. SCHENTHAL, J. E., SWEENEY, J. W., AND NETTLETON, W., JR. Clinical application of large-scale electronic data processing apparatus. *J. Amer. Med. Assoc.* (May 7, 1960).
4. SCHWICHTENBERG, A. H., FLICKINGER, D. D., AND LOVELACE, R. W. Development and use of medical machine record cards in astronaut selection. *U. S. Armed Forced Med. J.* **10** (1959), 11.
5. CECIL, R. *Textbook of Medicine.* Saunders & Co. (1959).
6. HARRISON, T. *Internal Medicine.* McGraw-Hill (1962).

ROBERT W. FLOYD—continued

may be proven unambiguous, the algorithm will eventually do so. If it is ambiguous, the algorithm will eventually generate the same string twice. Since no algorithm can decide the ambiguity of every phrase structure grammar, there must be grammars which are unambiguous, but for which no proof exists of freedom from ambiguity.

REFERENCES

1. NAUR, PETER, ET AL. Report on the algorithmic language ALGOL 60. *Comm. ACM 3*, 5 (May 1960), 299–314.
2. CHOMSKY, N. On certain formal properties of grammars. *Information and Control 2* (1959), 137–167; A note on phrase structure grammars. *Information and Control 2* (1959), 393–395.
3. BAR-HILLEL, Y., PERLES, M., AND SHAMIR, E. On formal properties of simple phrase structure grammars. *Zeit. Phonetik, Sprachwissenschaft und Kommunikationsforschung 14* (1961), 143–172.
4. FLOYD, R. W. A note on mathematical induction on phrase structure grammars. *Information and Control 4* (1961), 353–358.
5. POST, E. Avariant of a recursively unsolvable problem. *Bull. Amer. Math. Soc. 52* (1946), 264–268.

OFFICIAL NOTICES—cont'd from page 497

1963. "Due to the growth in exhibitors and attendance at the JCC's it has become necessary for us to look for more space than conventional conference facilities offer," Willis Ware has explained. "We feel it is to the advantage of both attendees and exhibitors that we keep all activities under one roof and have not wanted to sacrifice this arrangement. Considerable time and effort has been spent by our committee in finding facilities that maintain the quality of arrangement we feel important."

The main auditorium at the Las Vegas Convention Center seats 8500 and the center includes 90,000 sq. ft. of exhibit space and 17 meeting rooms for numbers from 25 to 1000. It is one of the few centers in the world specifically designed for conference and exhibition activities.