Summary of the Issue

THE ISSUE begins with an invited paper, "The Syntax of Programming Languages—A Survey," by Robert W. Floyd, covering an area of programming language which has begun to yield to mathematical treatment. A programming language to be useful not only must be a good vehicle for expression, but also must be such that a computer can analyze and translate source text. Floyd's survey describes the current state of the art in the application of formal mathematical methods to the study of programming languages. This treatment is augmented by a contributed paper, "Decision Problems of Phase Structure Grammars," by Peter S. Landweber. An application of the methods developed by investigators into formal structure is provided through Russell A. Kirsch's paper, "Computer Interpretation of English Text and Picture Patterns."

A significant portion of engineering-type calculations and other programs are written today using FORTRAN as a programming language. It enjoys international usage and is currently under active consideration for both national and international standardization. ALGOL is a more recent development which represents the principal contender to FORTRAN for adoption as a standard. Invited papers, "ALGOL," by Michael Woodger, and "FORTRAN," by John Backus and William P. Heising, trace the history of development of those two languages and impart something of their flavor.

Besides the languages which are intended primarily for the statement of computational algorithms, there have come into being some which are intended to state the solutions to problems in diverse areas of application but which are more specialized in the computer processes they control. The paper, "Introduction to FORMAC" by Jean E. Sammet and Elaine R. Bond, describes a major adjunct to the FORTRAN language which permits the computer user to manipulate algebraic expressions in a formal way. The paper, "List Processing and Extension of Language Facility by Embedding," by Daniel G. Bobrow and Joseph Weizenbaum, discusses a method of extension of the power of list processing. The two papers by Donald E. Knuth and John L. McNeley, "SOL—A Symbolic Language for General-Purpose Systems Simulation," and "A Formal Definition of SOL," are concerned with a system for modeling and studying systems behavior.

Several papers in this issue describe language developments intended to enhance communication between the practicing engineer and the computer. The distinction is by no means sharp between this category and the developments cited above. Perhaps it is the orientation of the language designer that gives each class of effort its distinctive character. In the one case, the language designers were motivated to produce a tool of general utility. In the other, the designer of the language was already involved in a particular area of application and developed a language for the solution of problems in that and related areas. Whether correctly classified or not, this editor places in the next group the papers: "A Versatile Problem-Oriented Language for Engineers," by Roger A. Gaskill; "A Logic Design Translator Experiment Demonstrating Relationships of Language to Systems and Logic Design," by Roy M. Proctor; "Aspects of Language Design for Combinatorial Computing," by Mark B. Wells; and "A Formal Language for Describing Machine Logic, Timing, and Sequencing (LOTIS)," by H. P. Schlaeppi.

Finally, there are two papers which do not fit easily in any of the three preceding categories. "A Procedure-Oriented Machine Language," by A. P. Mullery, describes investigations into the structure of internal addressing from a programmer's viewpoint, and the paper by J. T. Carleton, P. E. Lego, and R. M. Suarez, "A FORTRAN Extension to Facilitate Proposal Preparation," illustrates an application of FORTRAN that should be of special interest to those concerned with one of the problems of administering a technical center.