puter to an instruction not in that sequence. Safeguards can be provided to guarantee the return of control to the original sequence; and, on balance, these safeguards make the operations more powerful. The **Execute** mode of instruction sequencing has many uses in subroutine linkages, in special programming devices, and in monitoring routines.

REFERENCES

1. JULES MERSEL, "Program Interruption on the Univac Scientific Computer," *Proceedings of the WJCC*, p. 52 (1956).
2. F. P. BROOKS, "A Program-Controlled Program Interruption System," *Proceedings of the EJCC*, p. 128 (1957).
3. Reference Manual, IBM 709 Data Processing System, p. 37 (1958).
4. YU. A. MAKHMUDOV, *Radioteknika 3* (Mar., 1959) 44–57. In English: *Comm. ACM 2*, No. 10 (Oct. 1959), 3.

An Algorithm Defining ALGOL Assignment Statements

ROBERT W. FLOYD, *Armour Research Foundation, Illinois Institute of Technology, Chicago, Ill.*

It is not possible, by testing symbol pairs only [2], to determine whether a given symbol string is consistent with the formation rules of ALGOL [1]. For example, the formula

$$l1:\ l2:\ x[i := 5j + 3.14.159;$$

violates four distinct formation rules of ALGOL, yet each pair of adjacent characters may appear in permissible formulae. The algorithm described here will determine, with minor restrictions, whether a particular symbol string is a permissible ALGOL assignment statement. I believe that the same technique may be extended to determine whether a given symbol string is a permissible ALGOL program or not, where a program is defined as a sequence of permissible statements separated by semicolons. The algorithm scans the formula from left to right, replacing certain character pairs by single characters. If under the allowable transformations the symbol string may be reduced to the single character Σ, it is a well-formed formula in ALGOL; otherwise it violates the formation rules.

The algorithm represented by figure 1 assumes that the formula is stored character by character in an array S_i, beginning with S_1. A second array R_j is used for temporary storage. The generalization function $G(S_i)$ is used to replace all letters by I and all digits by G. Unambiguous binary operators such as $\times$, $/$, and $\uparrow$ [3] are replaced by $\odot_b$. $+$ and $-$, which are ambiguous in that they may serve either as unary or as binary operators, are replaced by $\odot_a$. For other characters S_i, $G(S_i) = S_i$.

Table 1 describes the tabular function M, mapping each character pair into either zero or some single character. Where no entry is made in table 1, the value of M is taken to be zero. It is assumed that each character has a nonzero machine representation.

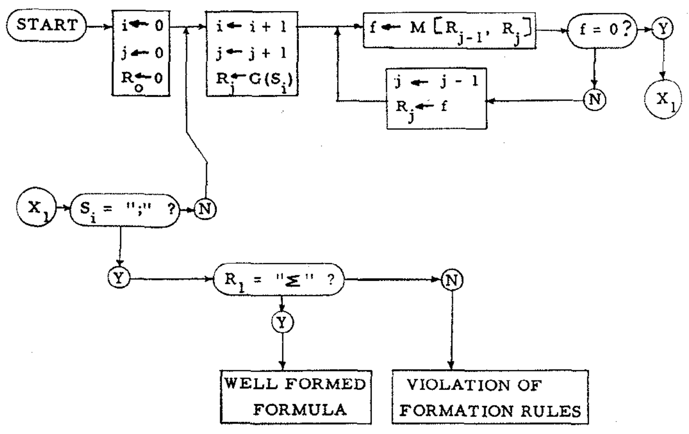

FIG. 1

TABLE 1

	R_j															
R_{j-1}	.	10	G	I	S	E	$\odot_a$	$\odot_b$	)	]	,	;	E_{rp}	E_{rb}	E_t	RH
.			N													
10			N*				N_2									
N_1			N*				N_2									
N_2			N*													
G	N	N_1	G				$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				
N		N_1	N				$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				
N*			N*				$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				
I			I	I	V	F	$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				Σ
V							$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				Σ
E							$\odot_u$	$\odot_u$	E_{rp}	E_{rb}	E_c	E_t				
$\odot_u$													E_{rp}	E_{rb}	E_t	
$\odot_a$													E_{rp}	E_{rb}	E_t	
(													E			
[														S		
E_c													E_{rp}	E_{rb}		
:=															RH	

The characters introduced by the substitution process have the following meanings:

G	an integer
N	a number containing a decimal point
N_1	an incomplete number, ending in $_{10}$
N_2	an incomplete number, ending in $_{10}\pm$
N^*	a number ending with an exponent of 10
I	an identifier; a letter followed by letters or digits
V	a subscripted variable
E	a parenthesized expression
S	a bracketed subscript
$\odot_u$	a unary arithmetic operator
$\odot_b$	a binary operator
$\odot_a$	an ambiguous operator (+ or −), unary or binary according to context
E_c	an expression followed by a comma
E_{rp}	an expression followed by a right parenthesis
E_{rb}	an expression (or list of expressions separated by commas) followed by a right bracket
E_t	an expression followed by a semicolon
RH	the replacement operator := followed by E_t
Σ	an identifier or subscripted variable followed by RH; a well-formed formula
F	a function

REFERENCES

1. Preliminary Report—International Algebraic Language, *Communications of the ACM 1* (Dec. 1958), 8–22.
2. *Communications of the ACM 2* (April 1959), 10–11.
3. Recommendations of the SHARE ALGOL Committee, *Communications of the ACM 2* (October 1959), 25–26.

Numerical Inversion of Laplace Transforms*

LOUIS A. SCHMITTROTH, *Phillips Petroleum Co., Idaho Falls, Idaho*

1. Introduction

This note describes a method for computing the inverse of a Laplace transform $F(s)$, when it is known that all singularities of $F(s)$ lie in the left half-plane, $\text{Im}(s) < 0$. The method has been programmed for the IBM 650 and satisfactory results obtained. Some limitations and possible extensions will be indicated below.

The impetus for the development of the program came from a problem in the design of a reactor control system. The control system under consideration uses two control loops, one of which has two time delays, so that the resulting transfer function is of a complicated type involving exponentials in a nontrivial manner. It seemed computationally prohibitive to try the traditional approach of poles and residues, so the present direct method was developed.

* Work done under contract to the U. S. Atomic Energy Commission.

2. The Complex Inversion Integral

If a given function $F(s)$ fails to fall into a table of Laplace transforms, the usual procedure is to try to invert it by use of the complex inversion integral:

$$I(t) = \frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F(s)e^{st}\, ds. \tag{1}$$

Here c is any real constant such that all singularities of $F(s)$ are in $\text{Im}(s) < c$.

It is assumed that $F(s)$ has an inverse $f(t)$ (continuous and of exponential order) and that the inversion integral represents $f(t)$ in the sense that (see Churchill [1], Ch. 6):

$$\frac{1}{2\pi i} \int_{c-i\infty}^{c+i\infty} F(s)e^{st}\, ds = \begin{cases} 0, & t < 0, \\ \frac{1}{2}f(0+), & t = 0, \\ f(t), & t > 0. \end{cases} \tag{2}$$