THE UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Department of Philosophy

Technical Report

SEQUENCE GENERATORS AND FORMAL LANGUAGES

Arthur W. Burks
Jesse B. Wright

ORA Project 03105

under contract with:

DEPARTMENT OF THE NAVY
OFFICE OF NAVAL RESEARCH
CONTRACT NO. Nonr 1224(21)
WASHINGTON, D.C.

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

September 1961

Engn
UMR 0836

Most of the results presented herein are drawn from our much longer paper which is to be published in the Proceedings of the Symposium on Recursive Function Theory held in New York City, April 6-7, 1961, and sponsored by the American Mathematical Society, the Association for Computing Machinery, and the Association for Symbolic Logic. The terminology and mode of development in the present paper differ considerably from that of the longer paper.

The fundamental concept is that of a sequence generator. An sequence generator is a finite, directed, labeled graph. Each node may or may not be labeled as a root R. Each node is labeled with a pair of truth values.

We are particularly interested in what we call the "behavior" of a sequence generator (see Fig. 2). The behavior of a sequence generator is a set of matrices of truth values. The pairs of truth values taken from the nodes along an infinite path of the graph constitute a two-by-omega matrix. Such a two-by-omega matrix belongs to the behavior of a sequence generator if and only if it is taken from a path of the graph which begins with a root. For example, the first element of Fig. 2 is taken from the path that starts with the root and always stays in the root, while the second element is taken from the path that starts with the root and oscillates back and forth between it and the other node.

This concept of behavior may be connected to the ordinary one by calling the upper truth value of a node an input and calling the lower truth value of a node an output. A sequence generator then determines a relation between infinite input sequences and infinite output sequences, which relation constitutes its behavior.

The inputs and outputs labeling nodes may be vectors of truth values rather than single truth values. Our results all apply to this more general case, but in the interest of simplicity we will present them in terms of the case where the input and output vectors are of length one.

Digital computers or finite automata are closely related to a special type of sequence generator called a deterministic sequence generator. Figure 3 is deterministic. A sequence generator is deterministic when it satisfies these two conditions:

(1) For each truth value, there is exactly one root having that as its input truth value.

(2) For each node and each truth value, there is exactly one node following the given node and having the given truth value as its input.

The deterministic sequence generator of Fig. 3 corresponds, in fact, to a binary counter.

Consider a graph derived from a digital computer or finite automaton in the following way. Each pair consisting of an input state and an internal state is a complete state. A node of the graph is provided for each complete state. The labels on a node give the input state and output state associated with the corresponding complete state of the automaton. Directed arrows of the graph indicate direct transitions between complete states. Any graph derived from a digital computer in this way is a deterministic sequence generator, and each deterministic sequence generator corresponds to a digital computer.

The rows of truth values in the infinite matrices introduced earlier define predicates on the natural numbers. This suggests the use of logical formulas to represent sequence generators. We do this in an interpreted system called the sequential calculus.

A formula of the sequential calculus is given in Fig. 4. The symbols of the sequential calculus are: individual variables, interpreted as ranging over the natural numbers; monadic predicate variables, interpreted as ranging over predicates of natural numbers; the individual constant zero; the successor function; truth-functional connectives; and quantifiers on both individual and predicate variables.

To each sequence generator there corresponds a formula of the sequential calculus, as illustrated in Fig. 4. The existential predicate quantifier "there is an s" asserts the existence of an infinite sequence of nodes satisfying four conditions: $s(o)$ says that this infinite sequence of nodes begins with the root R, $(\forall\tau)[s(\tau) \vee s(\tau')]$ that it is indeed a path of the graph; $i(\tau) \equiv s(\tau)$ and $o(\tau) \equiv \overline{s(\tau)}$ define the input and output predicates for the path. This correspondence between formulas and sequence generators is such that a pair of predicates of natural numbers $<i,o>$ satisfies the formula if and only if the corresponding two-by-omega matrix of truth values belongs to the behavior of the given sequence generator.

See next the normal form of Fig. 5. Every formula derived from a sequence generator can be put in the <u>normal form</u> $(\exists s)[R\&((\forall\tau)P)]$, where R and P are both truth functions, the root condition R being a truth function of $s(o)$, P being a truth function of $s(\tau)$, $s(\tau')$, $i(\tau)$, and $o(\tau)$. Conversely, one can derive from any normal form formula a corresponding sequence generator.

Thus formulas of the sequential calculus can be studied by means of sequence generators, to which combinatorial methods can be easily applied. We will next present a theorem which was arrived at by working with sequence generators but is more easily stated here as a theorem about formulas of the sequential calculus.

We introduce the following definition for a formula $\mathcal{C}(i,o)$ containing i and o as the only free variables. $\mathcal{C}(i,o)$ is <u>uniquely solvable</u> if and only if for each predicate i there is <u>exactly one</u> predicate o such that the pair

$\langle i,o \rangle$ satisfies $\mathcal{C}(i,o)$ in the intended interpretation. It is clear that when a formula $\mathcal{C}(i,o)$ is uniquely solvable, it *implicitly* defines a function $\mathcal{H}$ such that $o = \mathcal{H}(i)$ if and only if $\mathcal{C}(i,o)$. $\mathcal{H}$ is a function which maps the set of all predicates of natural numbers into the set of all predicates of natural numbers.

We are interested in a recursive definition of the type Church calls an unrestricted singulary recursion. As Fig. 6 shows, such a recursion gives:

$s(o)$ as a truth function of $i(o),\ldots, i(h)$

$s(\tau')$ as a truth function of $s(\tau)$; $i(\tau'),\ldots, i(\tau'+h)$

$o(\tau)$ as a truth function of $s(\tau)$,

where h is a constant.

Our *first theorem* is a definability theorem for uniquely solvable normal form formulas. The theorem is: for every uniquely solvable normal form formula $\mathcal{C}(i,o)$, there exists an equivalent formula $(\exists s)(\tau)\mathcal{U}(s,i,o,\tau)$, where $\mathcal{U}(s,i,o,\tau)$ is an unrestricted singulary recursion. Furthermore, the formula $(\exists s)(\tau)\mathcal{U}(s,i,o,\tau)$ can be constructed effectively from the given formula $\mathcal{C}(i,o)$.

The proof of the definability theorem has three steps. The first can be expressed if we think of the natural numbers as discrete times with τ being a time variable. Consider now a uniquely solvable normal form formula $\mathcal{C}(i,o)$ and the corresponding sequence generator with input i and output o. The output o at time τ may, contrary to physical reality, depend on input states which occur at a time later than τ. Let k be the number of nodes of the sequence generator. It turns out that the amount of anticipation is bounded by k^2, that is, the parameter h of the unrestricted singulary recursion is in fact k^2. Thus the first step of the proof of the definability theorem establishes the fact that for a uniquely solvable sequence generator with k nodes, the output o at time τ is independent of input states after $\tau+k^2$.

The second step of the proof of the definability theorem involves a function or operation on the class of sequence generators, called the subset sequence generator operation, and used by Myhill, Medvedev, Rabin, and Scott. Let Γ be a sequence generator and let Γ^* be *the subset sequence generator of* Γ. The nodes of Γ^* are sets of nodes of Γ. The arrows of Γ^* are placed according to the arrows of Γ in such a way as to take account of the fact that a binary relation induces on the subsets of its domain a function which is single-valued. There are two important facts about Γ^*. First, if Γ is uniquely solvable, then for each predicate i there is exactly one path of Γ^* giving rise to i. Second, Γ^* has the same behavior as Γ. Previous users of the subset sequence generator operation have shown this second fact to be so when behavior is based on finite sequences. This behavioral equivalence may be extended to our concept of behavior, which is based on infinite sequences, by means of König's infinity lemma.

As a consequence of steps one and two, our consideration of the normal form formula $\mathcal{C}(i,o)$ is reduced to a sequence generator Γ^* and a number k^2 having these two properties: first, for each predicate i there is a unique path in Γ^* giving rise to i; second, k^2 is a bound on the time dependence of the predicates s and o on the predicate i, since $s(\tau)$ and hence $o(\tau)$ are independent of the input states after $\tau+k^2$. To obtain the unrestricted singulary recursion for $\mathcal{C}(i,o)$ we must express $s(\tau')$ as a truth function of $s(\tau)$ and $i(\tau),\ldots,i(\tau'+h)$. For any given i we use the predicate s defined by the corresponding path through Γ^* as the s of the recursion, and we take h to be k^2. A simple argument based on the uniqueness of the path will show that $s(\tau')$ is a time-independent truth function of $s(\tau)$ and $i(\tau'),\ldots,i(\tau'+h)$. To write o as a restricted singulary recursion of i and s is now essentially a matter of an elaborate truth table.

Our definability result may be generalized to arbitrary formulas of the sequential calculus by means of some theorems about the sequential calculus established by J. R. Büchi. Büchi's main result is that there is a decision procedure for whether or not an arbitrary sentence of the sequential calculus is true in the intended interpretation.

An immediate corollary of this result is that there is a decision procedure for unique solvability, since an arbitrary formula $\mathcal{C}(i,o)$ is uniquely solvable if and only if the following formula of the sequential calculus is true:

$$(i)(\exists o)\left\{\mathcal{C}(i,o)\&(o_1)[\mathcal{C}(i,o)\supset(\tau)(o(\tau) \equiv o_1(\tau))]\right\}.$$

The generalization of our definability result involves the concept of a finitely anticipatory formula. A formula $\mathcal{C}(i,o)$ is <u>finitely</u> <u>anticipatory</u> if for each time τ there is a integer h such that the input states from time zero up to time $\tau+h$ determine the output state at time τ. It should be noted that in an unrestricted singulary recursion h is a constant, whereas in the definition of finitely anticipatory, h is a function of τ. Since $\leq$ can be defined in the sequential calculus, the statement that $\mathcal{C}(i.o)$ is finitely anticipatory can be expressed by a sentence of the sequential calculus. Therefore, the decidability of truth of sentences of the sequential calculus implies that the class of finitely anticipatory formulas is effectively decidable.

We can now state our general definability theorem. Let $\mathcal{C}(i,o)$ be any formula of the sequential calculus. The theorem states that the relationship between i and o can be expressed by an unrestricted singulary recursion if and only if $\mathcal{C}(i,o)$ is both uniquely solvable and finitely anticipatory. More precisely, let $\mathcal{C}(i,o)$ be any formula of the sequential calculus having i and o as the only free variables. There is an unrestricted singulary recursion $\mathcal{U}(s,i,o,\tau)$ such that $\mathcal{C}(i,o) \equiv (\exists s)(\tau)\mathcal{U}(s,i,o,\tau)$ if and only if $\mathcal{C}(i,o)$ is both uniquely solvable and finitely anticipatory. Moreover, there is an algorithm applicable to $\mathcal{C}(i,o)$ which yields an equivalent formula, $(\exists s)(\tau)\mathcal{U}(s,i,o,\tau)$, if one exists.*

The proof of the generalized definability theorem is too involved to be presented here, but we will make some remarks about it. The proof uses a normal form involving an infinite existential quantifier $(\exists^{\omega})$ (see Fig. 7). "$(\exists^{\omega})G$" means that there exist infinitely many natural numbers satisfying G. The infinite existential quantifier $(\exists^{\omega})$ is definable in the sequential calculus. Büchi has shown that, for any formula $\mathcal{C}(i,o)$ of the sequential calculus having the predicates i and o as the only free variables, there is an equivalent normal form formula. As shown in the top line of Fig. 7, this normal form is $(\exists s)[R\&(\exists^{\omega}\tau)G)\&((\forall\tau)P)]$, where R is a truth function of s(o), G is a truth function of $s(\tau)$, and P is a truth function of $s(\tau)$, $s(\tau')$, $i(\tau)$, and $o(\tau)$. This normal form is the same as the one we had before except for the addition of the conjunct $(\exists^{\omega}\tau)G$.

The generalized normal form of Fig. 7 corresponds to an extension of the idea of sequence generator in which each node may or may not be labeled as a goal G (see Fig. 8). The effect of the infinite quantifier $(\exists^{\omega})$ of the formula is obtained in a sequence generator with goals by considering only paths which pass through the goals infinitely many times. There is then a one-one correspondence between generalized normal form formulas and sequence generators with goals, which correspondence preserves behavior. Figure 8 gives an example of a sequence generator with goals and the equivalent generalized normal form formula.

It will be recalled that in the case of an ordinary uniquely solvable sequence generator with k nodes, the output o at time τ is determined by the input states from time zero up to time $\tau+k^2$. The same lemma holds for those sequence generators with goals which are both uniquely solvable and finitely anticipatory, so that, as before, the amount of anticipation is independent of time.

A complete presentation of the proof of the general definability theorem requires going inside Büchi's proof of decidability for the sequential calculus and making use of regular sets as defined by Kleene. We will not here present this proof but will conclude by summarizing the main result of this paper: there is an algorithm applicable to any formula of the sequential calculus with free input and output predicates. This algorithm produces an equivalent unrestricted singulary recursion, if one exists, and otherwise indicates that none exists.

*This theorem is not in the paper to be published in the Proceeding of the Symposium on Recursive Function Theory.

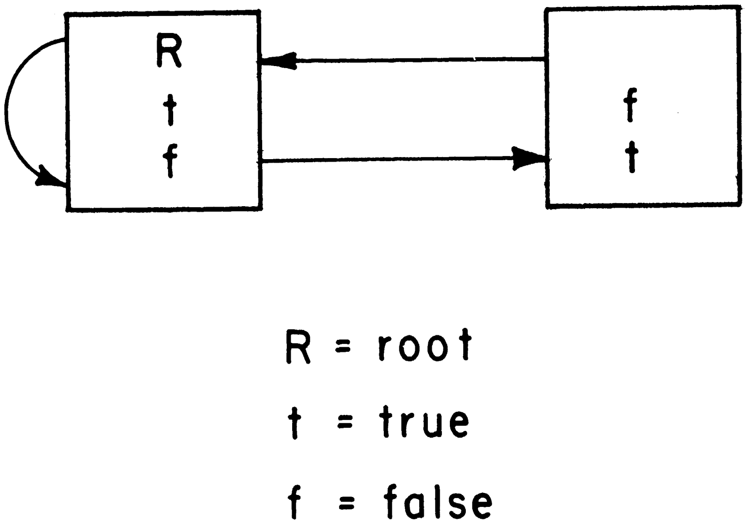


Fig. 1. Sequence Generator. Finite, directed, labeled graph.

```
R
| t t t t t t . . . |
| f f f f f f . . . |
R
| t f t f t f . . . |
| f t f t f t . . . |
```

Fig. 2. Two elements of the behavior of Fig. 1.

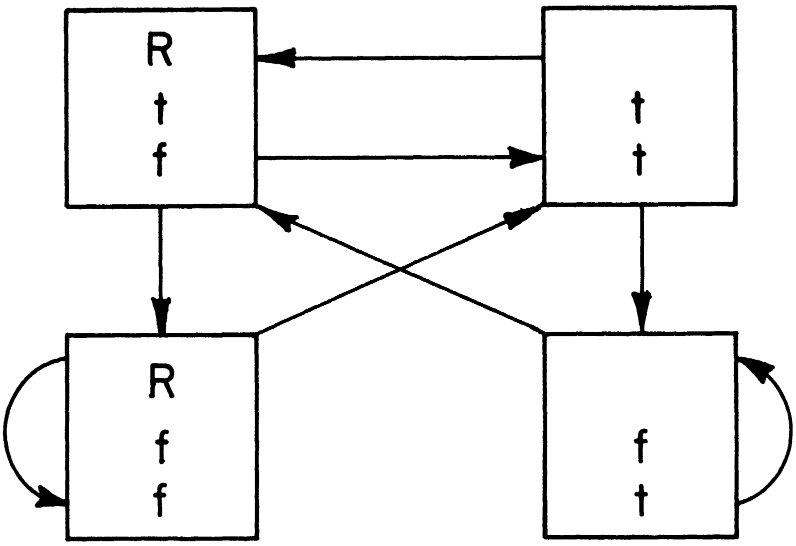
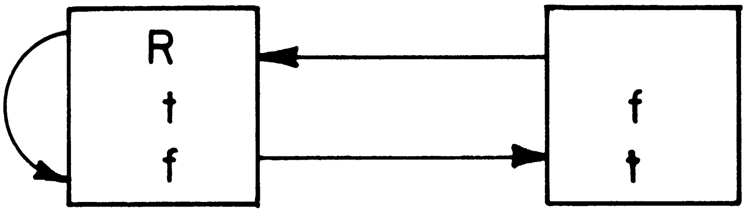


Fig. 3. Deterministic sequence generator for a binary counter.



$$
\begin{aligned}
&(\exists s)\ \{ s(o)\ \& \\
&\qquad (\forall \tau)\,[s(\tau) \vee s(\tau')]\ \& \\
&\qquad\qquad [i(\tau) \equiv s(\tau)]\ \& \\
&\qquad\qquad [o(\tau) \equiv \overline{s(\tau)}]\ \}
\end{aligned}
$$

Fig. 4. Sequence generator and formula of the sequential calculus.

$$
\begin{aligned}
&(\exists s)\ [R\ \&((\forall \tau)P)] \\
&R\ [s(o)] \\
&P\ [s(\tau),\ s(\tau'),\ i(\tau), o(\tau)]
\end{aligned}
$$

are truth functions

Fig. 5. Normal form.

$$
\begin{aligned}
&s(o) \equiv A\,[i(o), \ldots.\ i(h)]\ \& \\
&s(\tau') \equiv B\,[s(\tau);\ i(\tau'), \ldots.\ i(\tau' + h)]\ \& \\
&o(\tau) \equiv D\,[s(\tau)]
\end{aligned}
$$

A, B, D, are truth functions

Fig. 6. Unrestricted singulary recursion.

$$(\exists s)\left[R \,\&\, ((\exists^{\omega}\tau)\ G) \,\&\, ((\forall\tau)\ P)\right]$$

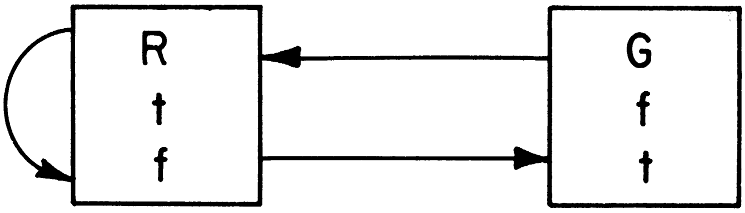
$$R\left[s(o)\right]$$

$$G\left[s(\tau)\right]$$

$$P\left[s(\tau),\ s(\tau'),\ i(\tau),\ o(\tau)\right]$$

are truth functions

Fig. 7. Generalized normal form.



$$(\exists s)\ \left\{s(o) \,\&\, (\exists^{\omega}\tau)\,\overline{s(\tau)} \,\&\, \right.$$
$$(\forall\tau)\left[s(\tau)\ \mathrm{v}\ s(\tau')\right] \,\&$$
$$\left.(i(\tau) \equiv s(\tau)) \,\&\, (o(\tau) \equiv \overline{s(\tau)})]\right\}$$

Fig. 8. Sequence generator with goals and equivalent generalized normal form formula.

BIBLIOGRAPHY

1. Büchi, J.R., "On a Decision Procedure in Restricted Second-Order Arithmetic," to appear in the Proceedings of the 1960 International Symposium on Logic, Methodology and Philosophy of Science, Stanford University Press.

2. Church, Alonzo, "Application of Recursive Arithmetic to the Problem of Circuit Synthesis." Summaries of talks presented at the Summer Institute for Symbolic Logic, Cornell Univ., 1957, Institute for Defense Analysis, Princeton, 1960.

3. Kleene, S.C., "Representation of Events in Nerve Nets and Finite Automata," in C.E. Shannon and J. McCarthy (eds.), Automata Studies, Princeton University Press, Princeton, N.J., 1956, pp. 3-41.

4. König, D., Theorie der Endlichen und unendlichen Graphen, Akademische Verlagsgesellschaft M.B.H., Leipzig, 1936.

5. Medvedev, I.T., "On a Class of Events Representable in a Finite Automaton," translated by J.J. Schorr-Kon from a supplement to the Russian translation of Automata Studies, C.E. Shannon and J. McCarthy (eds.), Group Report 34-73, Lincoln Laboratory, Lexington, Mass., 1958.

6. Myhill, J., "Finite Automata and Representation of Events," in Fundamental Concepts in the Theory of Systems, WADC Technical Report 57-624, ASTIA Document No. AD 1557 41, 1957.

7. Rabin, M.O., and D. Scott, "Finite Automata and Their Decision Problems," IBM Journal of Research and Development, 1959 3, 114-125.

DISTRIBUTION LIST
(One copy unless otherwise noted)

Asst. Sec. of Def. for Res. 2
and Eng.
Information Office Library Br.
Pentagon Bldg.
Washington 25, D. C.

Armed Services Tech. Infor. Agcy. 10
Arling Hall Sta.
Arlington 12, Virginia

Chief of Naval Research 2
Department of the Navy
Washington 25, D.C.
Attn Code 437, Inf. Syst. Br.

Chief of Naval Operations
OP-071-12
Navy Dept.
Washington 25, D. C.

Director, Naval Res. Lab. 6
Tech. Inf. Officer/Code 2000
Washington 25, D.C.

Commanding Officer, Officer of 10
Naval Research
Navy No. 100, Fleet P. O.
New York, New York

Commanding Officer, ONR Br. Office
346 Broadway
New York 13, New York

Commanding Officer, ONR Br. Office
495 Summer St.
Boston 10, Massachusetts

Office of Technical Services
Technical Reports Section
Department of Commerce
Washington 25, D.C.

Bureau of Ships
Department of the Navy
Washington 25, D.C.
Attn Code 671 NTDS

Chief, Bureau of Ships
Department of the Navy
Washington 25, D.C.
Attn Code 280

Chief, Bureau of Ships
Department of the Navy
Washington 25, D.C.
Attn Code 687E

Naval Ordnance Laboratory
White Oaks
Silver Spring 19, Maryland
Attn Technical Library

David Taylor Model Basin
Washington 7, D. C.
Attn Technical Library

Naval Electronics Laboratory
San Diego 52, California
Attn Technical Library

University of Illinois
Control Systems Laboratory
Urbana, Illinois
Attn D. Alpert

University of Illinois
Digital Computer Laboratory
Urbana, Illinois
Attn. Cr. J.E. Robertson

Technical Information Officer
U.S. Army Signal Research and Dev. Lab.
Fort Monmouth, New Jersey
Attn. Data Equipment Branch

DISTRIBUTION LIST (Continued)

Director 3
National Security Agency
Fort Geo. G. Meade, Maryland
Attn. Chief, REMP

Naval Proving Ground
Dahlgren, Virginia
Attn. Naval Ordn. Computation Center

National Bureau of Standards
Washington 25, D.C.
Attn. Dr. S.N. Alexander

Aberdeen Proving Ground, BRL
Aberdeen Proving Ground, Maryland
Attn. Chief, Computation Lab.

Office of Naval Research
Resident Representative
University of Michigan
820 E. Washington Street
Ann Arbor, Michigan

Commanding Officer
ONR, Branch Office
John Crerar Library Bldg.
86 East Randloph Street
Chicago 1, Illinois

Commanding Officer
ONR Branch Office
1030 E. Green Street
Pasadena, California

Commanding Officer
ONR Branch Office
1000 Geary Street
San Francisco 9, California

National Bureau of Standards
Washington 25, D.C.
Attn. Mr. R.D. Elbourn

Naval Ordnance Laboratory
Corona, California
Attn. H.H. Weider

George Washington University
Washington, D.C.
Attn. Prof. N. Grisamore

Dynamic Anal. and Control Lab.
Massachusetts Institute of Tech.
Cambridge, Massachusetts
Attn. D.W. Baumann

Burroughs Corporation
Research Center
Paoli, Pennsylvania
Attn. A.J. Meyerhoff

Hermes Incorporated
75 Cambridge Parkway
Cambridge 42, Massachusetts
Attn. Mr. Reuben Wasserman

Lockheed Missiles and Space Div.
3251 Hanover Street
Palo Alto, California
Attn. D.G. Willis

Univ. of Michigan
Ann Arbor, Michigan
Attn. Dept. of Philosophy,
Prof. A. W. Burks

Census Bureau
Washington 25, D. C.
Attn. Office of Asst. Director for
Statistical Services,
Mr. J. L. McPherson

National Science Foundation
Program Dir. for Documentation Res.
Washington 25, D. C.
Attn. Helen L. Brownson

DISTRIBUTION LIST (Continued)

Univ. of California - LA
Los Angeles 24, California
Attn Dept. of Engineering,
Prof. Gerald Estrin

Columbia University
New York 27, New York
Attn. Dept. of Physics,
Prof. L. Brilloutin

Hebrew University
Jerusalem, Israel
Attn. Prof. Y. Bar-Hillel

Massachusetts Inst. of Technology
Cambridge, Massachusetts
Attn. Prof. W. McCulloch

Benson-Lehner Corporation
1860 Franklin Street
Santa Monica, California
Attn Mr. Bernard Benson

Atomic Energy Commission
Washington 25, D. C.
Attn. Division of Research

Naval Research Laboratory
Washington 25, D. C.
Attn. Security Systems
Code 5266, Mr. G. Abraham

Cornell University
Department of Mathematics
Ithaca, New York
Attn. Prof. Mark Kac

Dr. A. M. Uttley
National Physical Laboratory
Teddington, Middlesex
England

Diamond Ordnance Fuze Laboratory
Attn. Library
Washington 25, D. C.

U.S. Army Signal Research and Dev. Lab.
Fort Monmouth, New Jersey
Attn. M. Tenzer

Harvard University
Cambridge, Massachusetts
Attn School of Applied Science,
Dean Harvey Brook

The University of Chicago
Institute for Computer Research
Chicago 37, Illinois
Attn. Mr. Nicholas C. Metropolis, Dir.

Commander
Wright Air Development Division
Wright Patterson Air Force Base, Ohio
Attn. WCLJR, Maj. L. M. Butsch

Laboratory for Electronics, Inc.
1079 Commonwealth Ave.
Boston 15, Massachusetts
Attn. Cr. H. Fuller

Stanford Research Institute
Computer Laboratory
Menlo Park, California
Attn. H. D. Crane

General Electric Co.
Schnectady 5, New York
Attn. Library, L.M.E. Dept.,
Bldg. 28-501

The Rand Corporation
1700 Main Street
Santa Monica, California
Attn. Numerical Analysis Dept.
Willis H. Ware

DISTRIBUTION LIST (Continued)

Hunter College
New York 21, New York
Attn. Dean Mina Rees

General Electric Research Laboratory
P.O. Box 1088
Schenectady, New York
Attn. Information Studies Section
R. L. Shuey, Manager

Radio Corporation of America
Moorestown, New Jersey
Attn. Missile and Surface Radar Div.
Sidney Kaplan

University of Pennsylvania
Institute of Co-Operative Research
Philadelphia, Pennsylvania
Attn. Dr. John O'Conner

Stanford Research Institute
Menlo Park, California
Attn Dr. Charles Rosen
Applied Physics Group

Northeastern University
360 Huntington Avenue
Boston, Massachusetts
Attn. Prof. L. O. Dolansky

Marquardt Aircraft Company
16555 Saticoy Street
P. O. Box 2013 - South Annex
Van Nuys, California
Attn. Dr. Basun Chang, Res. Scientist

Texas Technological College
Lubbock, Texas
Attn. Paul G. Griffith
Dept. of Electrical Engineering

IBM Corporation
Military Products Division
Owego, New York
Attn. Dr. S. Winkler

Post Office Department
Office of Research and Engineering
12th and Pennsylvania Ave.
Washington 25, D. C.
Attn. Mr. R. Kopp, Res. and Dev. Div.

Air Force Cambridge Res. Center
L. G. Hanscom Field, Bedford, Mass.
Attn. Chief, CRRB

Office of Chief Signal Officer
Department of the Army
Washington, D. C.
Attn. Res. and Dev. Div. SIGRO-6D
Mr. L. H. Geiger

Bell Telephone Laboratories
Murray Hill Laboratory
Murray Hill, New Jersey
Attn. Dr. Edward F. Moore

National Biomedical Research Inst.
9301 19th Avenue
Hyattsville, Maryland
Attn. Dr. R. S. Ledley

National Bureau of Standards
Washington 25, D. C.
Attn. Mrs. Frances Neeland

University of Pennsylvania
Moore School of Electrical Eng.
200 South 33rd Street
Philadelphia 4, Pennsylvania
Attn. Miss Anna Louise Campion

Varo Manufacturing Company
2201 Walnut Street
Garland, Texas
Attn. Fred P. Granger, Jr.

Data Processing Systems Staff
Department of State
Washington 25, D. C.
Attn. F. P. Diblasi

DISTRIBUTION LIST (Continued)

Dr. Saul Gorn, Director
Computer Center
University of Pennsylvania
Philadelphia 4, Pennsylvania

Applied Physics Laboratory
Johns Hopkins University
8621 Georgia Avenue
Silver Spring, Maryland
Attn. Supervisor of Technical Reports

Bureau of Supplied and Accounts, Chief
Navy Department
Washington, D. C.
Attn. Cdr. J.C. Busby, Code W3

Auerbach Electronic Corporation
1634 Arch Street
Philadelphia 3, Pennsylvania

National Aeronautics and Space Admin.
Goddard Space Flight Center
Greenbelt, Maryland
Attn. Chief, Data Systems Division

Federal Aviation Agency
Bureau of Research and Development
Washington 25, D. C.
Attn. RD-375 / Mr. Harry Hayman

Mr. Donald F. Wilson
Code 5144
Naval Research Laboratory
Washington 25, D. C.

David Taylor Model Basin
Washington 7, D. C.
Attn. Aerodynamics Lab., Code 628
Miss Cravens

Chief, Bureau of Ships
Code 671A2
Washington, D. C.
Attn. Lcdr. E. B. Mahinske, USN

Lincoln Laboratory
Massachusettes Inst. of Technology
Lexington 73, Massachusettes
Attn. Library