

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

Büchi, J. Richard

Final Report

RESEARCH IN AUTOMATA STRUCTURE, BEHAVIOR, AND DESIGN

J. R. Büchi
C. C. Elgot
J. H. Holland

UMRI Project 2794

under contract with:

DEPARTMENT OF THE ARMY
SIGNAL SUPPLY AGENCY
CONTRACT NO. DA 36-039-SC-78057
FORT MONMOUTH, NEW JERSEY

administered by:

THE UNIVERSITY OF MICHIGAN RESEARCH INSTITUTE ANN ARBOR

November 1960

ENGN
UMR 1100

Other agencies of the Federal Government contributed to the support of this research: the Office of Naval Research, the Army Office of Ordnance Research, and the National Science Foundation.

INTRODUCTION

Research under Contract No. DA-36-039-SC-78057 began in July, 1958, and ended in October, 1960. Eight Quarterly Summary Reports and seven Technical Reports were prepared and distributed to the U.S. Army Signal Corps. The interested reader is referred to the Technical Reports for the details of the research. These reports are as follows:

1. Decision Problems of Finite Automata Design and Related Arithmetics, by Calvin C. Elgot, June, 1959, 50 pp.
2. Cycles in Logical Nets, by John H. Holland, August, 1959, 61 pp.
3. Weak Second-Order Arithmetic and Finite Automata, by J. Richard Büchi, September, 1959, 45 pp.
4. Regular Canonical Systems and Finite Automata, by J. Richard Büchi, December, 1959, 25 pp.
5. A Universal Computer Capable of Executing an Arbitrary Number of Sub-Programs Simultaneously, by John H. Holland, February, 1960, 22 pp.
6. Iterative Circuit Computers, by John H. Holland, May, 1960, 14 pp.
7. On a Problem of Tarski, by J. Richard Büchi, July, 1960, 8 pp.

Four major automata research areas were investigated:

- (A) Automata design algorithms in formal systems of logic (explicated principally in items 1, 3, and 7 above).
- (B) Extensions and alternative formulations of Kleene automata theory (explicated principally in items 1, 3, and 4 above).
- (C) The role of "cycle-complexity" in nets and finite automata (item 2 above).
- (D) Systems of theoretical computers with iterative structure and highly parallel action (items 5 and 6 above).

A. AUTOMATA DESIGN ALGORITHMS IN FORMAL SYSTEMS OF LOGIC

Formal languages, such as the various predicate calculi, can be employed to set down expressions for the structure and the behavior of automata. A formal language is "suitable" for this purpose if it has within its "vocabulary" means by which (at a minimum) junctions, states of junctions at various times, and relations between junctions can be expressed. In general, the larger the "vocabulary," the easier it is for the designer to set down the automata conditions he wishes to have satisfied.

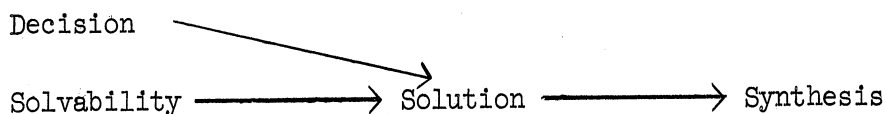
Explication of the exact relationships between expressions for automata behavior and expressions for automata structure is important for the theory of the mechanical design of computing machinery. Several important questions arise: (1) Is there an algorithm such that, given an automaton behavior condition expression, it can be determined whether there is an expression for an automaton satisfying the condition? (Solvability Problem.) (2) Is there an algorithm such that, given expressions for an automaton behavior condition and for an automaton structure, it can be determined whether the automaton satisfies the condition? (Solution Problem.) (3) Is there an algorithm such that, given a behavior condition, an automaton satisfying the condition can be produced (providing such an automaton exists)? (Synthesis Problem.)

If these three questions could be answered affirmatively, the process of the design of computing machinery could be much improved. For instance, a designer might begin by setting down some condition he wishes a computer circuit to satisfy. The Solvability result would tell him whether the condition can be satisfied by an automaton. If the answer is "yes," the designer could apply the Synthesis result to the condition and obtain the specifications for an automaton satisfying the desired condition.

If the designer had a condition to be satisfied and also a tentative automaton solution, he might apply the Solution result, to determine whether indeed the automaton does satisfy the condition. It can be shown that the existence of the Solvability result implies the existence of the Solution result, which in turn implies the existence of the Synthesis result [Büchi, Elgot, and Wright, "The Non-Existence of Certain Algorithms of Finite Automata Theory," Abstract: Notices of the American Mathematical Society, 5, 98 (February, 1958)].

This question of the existence of the Solvability, Solution, and Synthesis results can be shown to be bound up with the question of the existence of a decision procedure for the formal language in which the automata and automata condition are expressed. (A decision procedure is a mechanical means of determining the truth or falsity of the expressions of the language.) It can be shown that the existence of a Decision procedure for the formal language implies the existence of the Solution result (Büchi, Elgot, and Wright, op. cit.).

Diagrammatically expressed, the algorithms are related in the following fashion.



Thus, the Decision and the Solvability results are the "strongest," while the Synthesis result is the "weakest."

One attack on the problem of the mechanical design of computing machinery is thus seen to be the examination of "suitable" formal languages, ascertaining whether there is a decision procedure. If there is such a procedure, then the existence of Solution and Synthesis algorithms follows immediately. For many potentially useful formal languages, the question of the existence of a decision procedure is still open, and solution of this problem itself becomes a central goal of research.

Now, it is generally the case that the "stronger" a language (the larger its basic vocabulary and the broader the assertions that can be made in it) the more likely it is that there will be no purely mechanical method of deciding the truth or falsity of expressions in the language; such a language is said to be undecidable. Consequently, the researcher's problem becomes this: to show the existence of a decision procedure for the strongest possible language suitable for expressing automata and conditions on automata.

The record of the earlier research by the logic of Computers Group on automata languages and their decision procedures is contained in the two technical reports, Decision Problems of Finite Automata Design and Related Arithmetics, by Calvin C. Elgot (June, 1959), and Weak Second-Order Arithmetics and Finite Automata, by J. R. Büchi (Sept., 1959). In these papers, many potentially useful automata languages are considered and their properties evaluated. For several of these languages the desired algorithms can be shown to exist. Thus, in these papers it is shown that there are mechanical procedures for the design of some automata, for example, the so-called "Kleene behavior" automata along with some other classes of automata. Unfortunately, none of the languages considered is strong enough to allow general automata behavior conditions to be expressed, and, most unfortunately, conditions involving infinite automata behavior cannot be expressed. The search for a language which would allow general conditions on automata to be expressed, including conditions involving infinite behavior led to a consideration of the "Sequential Calculus." This language had been examined by A. Tarski and also by R. M. Robinson in his "Restricted Set-Theoretical Definitions in Arithmetic," Proc. Am. Math. Soc., 9, 238-242 (1958). The question of the existence of a decision procedure for the Sequential Calculus was open. This question was answered in the affirmative by J. R. Büchi in April of 1960 (see the Technical Note On a Problem of Tarski, July, 1960). From the existence of the decision procedure, the existence of the

Solution and Synthesis algorithms follows immediately.

The Sequential Calculus, interpreted for automata design, has the following "vocabulary":

1. Monadic predicate variable symbols denoting (arbitrary sets of natural numbers or) automata junctions.
2. Individual variable symbols denoting (natural numbers or) the times at which the junctions are in certain states.
3. An individual constant "0" standing (for the number zero or) for zero time.
4. The connectives ("and," "or," "not," "implies," "if-and-only-if") of propositional calculus.
5. The equality relation between individuals.
6. The "less than" relation between individuals.
7. The "successor" function to be applied to individual variables (and to the individual constant).
8. Quantifiers ("all," "for every") to be applied to both predicate and individual variables.

An example of a condition expressed in the Sequential Calculus is given below:

$$\begin{aligned}
 (\exists R) (\forall t) & \left\{ \left[S(t) \equiv [I(t) \ \& \ \sim J(t) \ \& \ \neg R(t)] \vee [\sim I(t) \ \& \ \sim J(t) \ \& \ R(t)] \vee \right. \right. \\
 & \left. [\sim I(t) \ \& \ J(t) \ \& \ \sim R(t)] \right\} \ \& \ \left\{ R(t) \equiv [I(t') \ \& \ \sim J(t') \ \& \ \sim R(t')] \vee \right. \\
 & \left. [\sim I(t') \ \& \ J(t') \ \& \ \neg R(t')] \vee [\sim I(t') \ \& \ \sim J(t') \ \& \ R(t')] \vee [I(t) \ \& \ J(t) \ \& \ R(t)] \right\} \ \& \\
 \exists(Q) (\forall t) & \left\{ [Q(t) \supset \sim I(t)] \ \& \ (\forall x) (\exists t) [x < t \ \& \ Q(t)] \right\}
 \end{aligned}$$

This example may be read (in part): "There is an automaton junction R such that for every instant of time t the output junction S at time t will be on if and only if the input junction I is on at t, and the input J is not on, and R is also not on,.....and there is a junction Q such that for every time t, if Q is on at t, then input I is not on, and, for every time x, there is a time t such that x is less than t and Q is on at time t."

Several important problems remain in this area of research.

1. Although the existence of a decision procedure for the Sequential Calculus implies the existence of Solution and Synthesis algorithms for automata design, the question of the existence of a Solvability algorithm for conditions expressed in the Sequential Calculus is still open.
2. The "vocabulary" of the Sequential Calculus should be extended by means of additional operations and relations defined in terms of the original primitives.
3. The usefulness of the Sequential Calculus may be extended by taking as individuals, not single numbers, but strings of numbers and replacing the successor function by an "initial segment" relation (viz. for strings x , y and z , x is an initial segment of y if and only if there is a z such that x followed by z is equal to y). The question of the decidability of this new language is open.

B. KLEENE AUTOMATA THEORY

In his paper "Representation of Events in Nerve Nets and Finite Automata" (RAND RM-704, 1951, and in Automata Studies, ed. Shannon and McCarthy, 1956), S. C. Kleene describes a class of finite automata with binary output. He shows that the classes of sequences which can be "detected" by such automata can be represented by "regular expressions." (An automaton is said to "detect" a sequence just in case the output is "on" after the sequence has been applied to the automaton input.) Kleene then shows how one can pass from a regular expression to an automaton (Synthesis theorem) and from an automaton to a regular expression (Analysis theorem).

Simplifications and extensions of Kleene's results have been provided by Copi, Elgot, and Wright ("Realization of Events by Logical Nets," JACM, 5, 181-199, 1958), John Myhill ("Finite Automata and the Representation of Events," WADC Report TR 57-624, Fundamental Concepts in the Theory of Systems, Oct., 1957), Rabin and Scott ("Finite Automata and Their Decision Problems," IBM Journal, April, 1959, 114-125), I. T. Medvedev (On A Class of Events Representable in a Finite Automaton, MIT Lincoln Laboratory Group Report, 34-73, trans. from the Russian by J. Schorr-Kon, June 30, 1958) and McNaughton and Yamada ("Regular Expressions and State Graphs for Automata," IRE, Trans. on Electronic Computers, Vol. EC-9, No. 1, March, 1960, pp. 39-48).

Although Kleene's Synthesis result provides a means by which a computer designer can obtain descriptions of finite automata satisfying desired automata behavior expressible in terms of regular expressions, the process is not entirely satisfactory. It is, for instance, difficult to set down, in terms of a regular expression, the behavior desired by an automaton.

In the research conducted by the Logic of Computers Group, an investigation has been made of alternative means of expressing the behavior of finite automata with binary output, that is, of expressing the sets which are defined by Kleene regular expressions.

Büchi and Elgot have investigated extensively a formal system of logic which closely resembles the Sequential Calculus described in the previous section. This language, a "weak second-order arithmetic," is identical to the Sequential Calculus except that the values that the predicate symbols may take on are restricted to finite sets of natural numbers (instead of arbitrary sets of natural numbers). Büchi and Elgot have shown that, for every formula of this language without free individual variables, there is a corresponding Kleene regular expression. (Decision Problems of Finite Automata Design and Related Arithmetics, by Calvin Elgot, June 19, 1959; Weak Second-Order Arithmetic and Finite Automata, by J. R. Büchi, September, 1959.)

Pursuing an alternative line of research, J. R. Büchi has shown (in Regular Canonical Systems and Finite Automata, Dec., 1959) that "Post production" systems can be employed to define precisely the set of sequences that is defined by Kleene regular expressions. Rules of the form $ax \rightarrow bx$ or $xa \rightarrow xb$, for instance, generate just the set of words detectable by a finite automaton. These results have suggested an investigation of production rules which are essentially stronger than these and thus describe the behavior of larger classes of automata.

C. NET CYCLE-COMPLEXITY

Cycles in an automaton have a profound effect on the automaton's behavior. It is not always simple, however, to see just what conditions the presence of cycles imposes on the behavior of an automaton. This sort of analysis problem for automata may be attacked by defining precisely the nature of the systems to be considered so that statements about automata (structure) logically imply statements about system behavior. John Holland (in Cycles in Logical Nets, August, 1959) uses the theory of nets and automata developed by Burks and Wright and Burks and Wang ["Theory of Logical Nets," Proc. IRE, 41, 1357-1365 (1953) and "The Logic of Automata," JACM, 4, 193-218, 279-297 (1957)] to provide precise definitions of systems with cycles. Several problems are then formulated and theorems proved.

1. Since a logical net with a periodic input sequence produces a periodic output sequence, can the spectrum of periodic outputs be related to the level of cycle complexity?
2. Is there a level of complexity such that any behavior possible for a fixed logical net can be realized by a logical net constructed only of cycles of this degree or less?

Holland provides partial answers to both these questions, showing the relation that exists between outputs and cycle complexity for nets having a particular cycle complexity, and showing that for nets with cycles of a particular sort there is no maximal degree of cycle complexity.

These notions of net and cycle complexity can be investigated further. One area of possible future research is the further ranking and classifying of finite automata by means of measures of cycle and net complexity.

D. ITERATIVE CIRCUIT COMPUTERS

There is an increasing need for improved computing machinery—machinery which is faster, more compact, lower in cost, more reliable, and which has larger capacity. There have been multiple attacks on these problems: increased pulse rate, greater use of parallel action and asynchrony, research into micro-circuitry and the machine manufacture of components. What also may be required is a radically different all-over computer design which can take maximum advantage of these separate advances.

John Holland, in A Universal Computer Capable of Executing an Arbitrary Number of Sub-Programs Simultaneously (Feb., 1960), proposes a computer having a 2-dimensional modular structure so that efficient use could be made of high element density and "template" techniques being developed in research on micro-miniature elements. Also, this proposed computer design would allow subprograms to be spatially organized and acted upon simultaneously. In addition, this "highly parallel" system can be made the basis for theoretical studies of machines that "grow" and "adapt."

More detailed consideration of the use of computers with iterative structure as the mathematical frame for research into growing automata is given in Iterative Circuit Computers (May, 1960), also by John Holland. It is indicated how the mathematical characterizations of classes of these iterative circuit computers might be employed in developing a theory of machine-adaptive systems.

DECISION PROBLEMS OF FINITE AUTOMATA DESIGN AND RELATED ARITHMETICS

(Technical Report 2755-6-T, June, 1959)

Calvin C. Elgot

Abstract

Certain formal arithmetics may be employed as design languages for finite automata design conditions, the notion of automaton, and the notion of an automaton satisfying a condition are expressible in these arithmetics. An automaton satisfies a condition if a certain formula of the arithmetic is valid.

For certain arithmetics, algorithms are produced which enable one to decide

- (1) whether a given automaton satisfies a given condition,
- (2) whether an automaton exists satisfying a given condition (and if there is one, producing one),
- (3) whether at most one automaton exists satisfying a given condition,
- (4) whether a given sentence is true.

These results make use of a theorem (5.3) which characterizes finite automata behavior by means of formulas of an arithmetic. The following corollary is typical of the side results obtained. If a natural number is identified with the set of natural numbers less than it, then the first-order theory of quasi-finite (finite or finite complement) sets of natural numbers based upon the Boolean set operations and the property of being a natural number is decidable.

For certain other arithmetics, it is shown that algorithms of the type indicated above fail to exist.

(This paper was printed and distributed as Technical Report 2755-6-T, June, 1959. It is to be published in the Transactions of the American Mathematical Society.)

CYCLES IN LOGICAL NETS

(Technical Report 2794-4-T, August, 1959)

John Holland

Abstract

The relation between the complexity of cycles in a logical net and the complexity of the behavior that results is investigated. Two problems are considered.

- (1) A logical net with a periodic input sequence produces a periodic output sequence; how is the spectrum of periodic outputs related to the level of cycle complexity?
- (2) Is there a level of complexity c such that any behavior possible for a fixed logical net can be realized by a logical net constructed only of cycles of complexity $c' \leq c$?

The first question is answered for the case of nets with cycles having a feedback coefficient $r = 1$. The second question is answered in the negative for individual cycles, and it is conjectured that this will be true for nets in general.

[This paper was printed and distributed as Technical Report 2794-4-T, August, 1959. It was published in the Journal of the Franklin Institute, 270, 202-226 (1960).]

WEAK SECOND-ORDER ARITHMETIC AND FINITE AUTOMATA

(Technical Report 2794-6-T, Sept., 1959)

J. Richard Büchi

Abstract

In essence, this paper states that a certain formal language, a weak second-order arithmetic (designated throughout this paper as W.2.A., and in Elgot, Decision Problems of Finite Automata Design and Related Arithmetics, as L_1^1) can be used in place of the formalism of regular expressions (developed by S. C. Kleene in his paper "Representation of Events in Nerve Nets and Finite Automata," in Automata Studies, Princeton University Press, 1956) in denoting the behavior of finite automata. The important Kleene Synthesis and Analysis theorems can also be obtained in this new formulation. (Synthesis: For every formula of W.2.A. one can construct an automaton with special output, such that the behavior of the automaton is just the set of predicates which satisfy the given formulas. Analysis: For every automaton with special output one can obtain a formula of W.2.A. such that the formula denotes the behavior of the automaton.) This result is of particular value because formulas of W.2.A. seem to be more convenient than regular expressions for formalizing conditions on the behavior of automata. Important additional results in pure logic are also obtained: the synthesis and analysis theorems yield valuable information on the strength of W.2.A. and related formalisms. [For expositions of Kleene's theory of regularity see Copi, Elgot, and Wright, "Realization of Events by Logical Nets," JACM, 5, 181-196 (1958); Rabin and Scott, "Finite Automata and Their Decision Problems," IBM Journal, 114-125 (April, 1959); and Myhill, "Finite Automata and Representation of Events," WADC Report TR 57-624, Fundamental Concepts in the Theory of Systems, October, 1957, pp. 112-137.]

(This paper was printed and distributed as Technical Report 2794-6-T, September, 1959. It was published in 1960 in the Zeitschrift für Mathematische Logik und Grundlagen der Mathematik, 6, 66-92.)

REGULAR CANONICAL SYSTEMS AND FINITE AUTOMATA

(Technical Report 2794-7-T, Dec., 1959)

J. Richard Büchi

Abstract

A special type of canonical system, called normal systems and containing rules of production of form $\underline{ax} \rightarrow \underline{xb}$ only, plays a role in Post's theory of effectiveness. One's attention is therefore naturally drawn to another type of canonical system, whose rules of production are of similar form, $\underline{ax} \rightarrow \underline{bx}$. We shall call these the regular systems.

Normal systems are of interest for their surprising strength; Post has shown that they produce all canonical (i.e., recursively generable) sets of words. Regular systems are interesting for the opposite reason. It is not difficult to see that they can produce only recursive sets of words, and therefore not even all recursive sets of words (diagonal argument). More surprising is our main result which says that only a very simple kind of set of words is generable by regular systems, namely, the regular sets or behaviors of finite automata.

To a finite automaton with binary output one can quite naturally adjoin a system Φ of regular productions and an axiom U in such a way that an input word x is detected by the output of the automaton just in case x is Φ -deducible from U , i.e., the set of words generated by Φ from U is the behavior of the finite automaton with binary output. Our main result is the following converse assertion. The set β of words generated by any regular system Σ from a finite set of axioms is the behavior of some finite automaton with binary output. Consequently, regular systems produce exactly the regular sets of words in the sense of Kleene; Copi, Elgot, and Wright; and Myhill.

(This paper was printed and distributed as Technical Report 2794-7-T, Dec., 1959. It was prepared for delivery to the Cambridge, Mass., meeting of the American Mathematical Society, October, 1959. The abstract appeared in the Notices of the American Mathematical Society, 6, Nov., 1959. Some of these results are also to be published in the Proceedings of the Berlin Mathematical Society.)

A UNIVERSAL COMPUTER CAPABLE OF EXECUTING
AN ARBITRARY NUMBER OF SUB-PROGRAMS SIMULTANEOUSLY

(Technical Report 2794-10-T, Feb., 1960)

John Holland

Abstract

This paper describes a universal computer capable of simultaneously executing an arbitrary number of sub-programs, the number of such sub-programs varying as a function of time under program control or as directed by input to the computer. Three features of the computer are:

- (1) The structure of the computer is a 2-dimensional modular (or iterative) network so that, if it were constructed, efficient use could be made of the high element density and "template" techniques now being considered in research on microminiature elements.
- (2) Sub-programs can be spatially organized and can act simultaneously, thus facilitating the simulation or direct control of "highly-parallel" systems with many points or parts interacting simultaneously (e.g., magneto-hydrodynamic problems or pattern recognition).
- (3) The computer's structure and behavior can, with simple generalizations, be formulated in a way that provides a formal basis for theoretical study of automata with changing structure (cf. the relation between Turing machines and computable numbers).

The computer presented here is one example of a broad class of universal computers which might be called universal iterative circuits. This class can be rigorously characterized and formally studied. The present formulation is intended as an abstract prototype which, if current component research is successful, could lead to a practical computer.

(This paper was printed and distributed as Technical Report 2794-10-T, Feb., 1960. It was published in the Proc. 1959 Eastern Joint Computer Conference.)

ITERATIVE CIRCUIT COMPUTERS

(Technical Report 2794-12-T, May, 1960)

John H. Holland

Abstract

A mathematical characterization is given of broad class of computers which are iterative or modular in structure (which allows efficient use of template techniques in construction), and which can process arbitrarily many words of stored data at the same time, each by a different sub-program if desired. This class of computer includes representatives structurally and behaviorally equivalent to Turing machines, "tessellation" automata of von Neumann and E. F. Moore, Burks and Wang "growing logical nets," and the "potentially infinite automata" of A. Church.

In this characterization, a computer can be considered to be composed of modules arranged in a 2-dimensional rectangular grid; the computer is homogeneous (or iterative) in the sense that each of the modules can be represented by the same fixed logical network. The modules are synchronously timed and time for the computer can be considered as occurring in discrete steps, $t = 0, 1, 2, \dots$

Basically each module consists of a binary storage register together with associated circuitry and some auxiliary registers. At each time-step a module may be either active or inactive. An active module, in effect, interprets the number in its storage register as an instruction and proceeds to execute it. There is no restriction (other than the size of the computer) on the number of active modules at any given time.

(This paper was prepared and distributed as Technical Report 2794-12-T, May, 1960. It was published in the Proceedings of the Western Joint Computer Conference, May, 1960.)

ON A PROBLEM OF TARSKI

(Technical Note 2794-13-T, July, 1960)

J. Richard Büchi

Abstract

I. On a Hierarchy of Monadic Predicate Quantifiers

Let x, x_1, x_2, \dots be variables ranging over natural numbers, and let i_1, i_2, i_3, \dots be variables ranging over monadic predicate (sets) of natural numbers, let $\underline{i}, \underline{j}$ denote k -tuples of i 's. The hierarchy $[\Sigma_m, \Pi_m]$ of predicates $P(\underline{i})$ on predicates is defined thus: $\Sigma_0 =$ all predicates $P(\underline{i})$ definable by formulas $K[\underline{i}(0)] \vee (\exists x) H[\underline{i}(x), \underline{i}(s+1)] \vee (\forall x) U[\underline{i}(x)]$, whereby K, H, U are truth functions in the indicated constituents, $\Pi_m = \{\sim P \mid P \in \Sigma_m\}$, $\Sigma_{m+1} = \{(\exists \underline{j}) P \mid P \in \Pi_m\}$. A k -tuple \underline{i} of monadic predicates may be considered to be an infinite sequence $\underline{i}(0) \underline{i}(1) \underline{i}(2), \dots$ whose elements are drawn from the finite alphabet consisting of all k -tuples of truth-values. Let $\underline{u}, \underline{v}, \dots$ denote finite words on this alphabet, let $\underline{u} \underline{v}$ denote concatenation. Def: $P(\underline{i})$ is of finite rank if there is a \sim -congruence relation $\underline{u} \sim \underline{v}$ on words, with finite partition, and such that $\underline{u} \underline{q} \sim \underline{v} \underline{q}$ implies $P(\underline{u}_1 \underline{u}_2 \dots) \equiv P(\underline{v}_1 \underline{v}_2 \dots)$. Theorem: The following are equivalent conditions on $P(\underline{i})$: (1) $P \in \Sigma_2$, (2) P is of finite rank, (3) $P = P_1 \dots P_a$ whereby each P_c is of the form $R \underbrace{S S S \dots}_n$, R and S being regular sets of finite words. This is established by using basic facts from automata-theory (regularity), the fan theorem (König's infinity lemma), and Ramsey's theorem [Proc. London Math. Soc., (2), 30, 264-286 (1930)]. Clearly P is of finite rank if and only if $\sim P$ is; therefore Corollary: $\Sigma_2 = \Pi_2$ and therefore $\Sigma_n = \Pi_n = \Sigma_2$ for $n \geq 2$.

II. On a Problem of Tarski

Let SC be the interpreted system containing variables x ranging over natural numbers, variables i ranging over monadic predicates on natural numbers, the successor function, propositional connectives, and quantifiers for both types of variables. Using the theorem and definition of the preceding section one obtains: Theorem 1: The predicates $P(i_1, \dots, i_k)$ definable in SC are exactly those belonging to Σ_2 . For example: exactly the ultimately periodic sets of natural numbers are definable in SC. Theorem 2: Truth of sentences in SC is decidable. This answers a question of Tarski [see R. M. Robinson, Proc. Am. Math. Soc., 9, 238-242 (1958)], and seems to be a rather strong result since

essential assertions about infinity can be stated in SC (parts of the fan-theorem and Ramsey's theorem). By interpreting predicates i as binary expansions of real numbers, theorem 2 may be stated thus: Theorem 2': The first-order theory of $[Re, +, Nn, Pw]$ is decidable. Here Re is the set of real numbers ≥ 0 , Nn is the set of natural numbers and Pw is the set of integral powers of 2. (Compare this result with Tarski's decidability of $> Re, +, .$.) Let SC_{per} be like SC , except that the variables i range over ultimately periodic predicates. Theorem 3: A sentence is true in SC_{per} if and only if it is true in SC . i.e., $[Re, +, Nn, Pw]$ and $[R, +, Nn, Pw]$ are arithmetically equivalent. Here R stands for the set of rational numbers.

[Two papers were prepared for the Missoula, Montana, meeting of the American Mathematical Society in June, 1960. The abstracts were published in the Notices of the American Mathematical Society, 7, 381-2 (June, 1960); the Technical Note presenting these results was printed and distributed in July, 1960. These results, under the title "A Decision Method for Restricted Set Theory," were presented at the International Congress for Logic, Methodology, and Philosophy of Science, Stanford, Cal., 1960, and will be published in the Proceedings of the Congress.]

UNITED STATES ARMY SIGNAL RESEARCH AND DEVELOPMENT LABORATORY
DISTRIBUTION LIST
CONTRACT NO. DA36-039 SC-78057

OASD (R and E) Room 3E1065
The Pentagon
Washington 25, D. C.
Attn: Technical Library

Chief of Research and Development
OCS, Department of the Army
Washington 25, D. C.

Chief of Research and Development
OCS, Department of the Army
Washington 25, D. C.
Attn: Dr. I. R. Hershner, Jr.

Director
U S Naval Research Laboratory
Washington 25, D. C.
Attn: Code 2027

Chief Signal Officer
Department of the Army
Washington 25, D. C.
Attn: SIGRD

Commanding Officer and Director
U S Navy Electronics Laboratory
San Diego 52, California

Director
U S National Bureau of Standards
Boulder Laboratories
Boulder, Colorado
Attn: Div 14-0, Library

Commander
Wright Air Development Center
Wright Patterson Air Force Base
Ohio
Attn: WCOSI-3

Commander
Air Force Cambridge Research Center
L. G. Hanscom Field
Bedford, Mass.
Attn: CROTLR-2

Commander 2
Rome Air Development Center
Griffiss Air Force Base
New York
Attn: RCSST-3

Commander 10
Armed Services Tech Info Agency
Arlington Hall Station
Arlington 12, Va.

Massachusetts Institute of Technology
Lincoln Laboratory
Cambridge 39, Massachusetts
Attn: W. Davenport, Jr.

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

Commanding Officer
Office of Ordnance Research
Box CM, Duke Station
Durham, N. C.

Commanding General
Ballistics Research Laboratory
Army Proving Ground, Maryland
Attn: Dr. C. V. L. Smith
Ch, Computation Lab.

Moore School of Electrical Engineering
University of Pennsylvania
220 South 33rd Street
Philadelphia 4, Pennsylvania
Attn: Prof. G. Patterson
Switching Theory Contract
Attn: Prof. S. Gorn

Army Mathematics Research Center 2
University of Wisconsin
Madison, Wisconsin
Attn: Dr. R. Langer

Commanding General 2
U S Army Signal School
Fort Monmouth, New Jersey

Commanding Officer
U S Army Signal R and D Laboratory
Fort Monmouth, New Jersey
Attn: NP (Dir Data Proc Fac Div) 2
Attn: NPT (Ch Data Transducer Br) 2
Attn: G (Mathematics Div.)

Mr. L. Leskowitz, Computation Ctr.)
Attn: XS (Mr. J. Borsuk, Explor Res Div S)
Attn: XS (Dir Explor Res Div S)
Attn: N (Dir Systems Engrg Div)
Attn: ADTE (Evans)

Attn: NR (Dir Trans Fac Div)
Attn: RHA (Records Holding Area)
Attn: TN (Tech Info - For: 3
Brit and Can Covts)
Attn: DR (Ofc Dir of Res Opns) 5

Staff
Stanford Electronics Laboratories
Stanford University
Stanford, California

This Report is distributed by Office of Research Operations, Fort Monmouth, New Jersey
Telephone: Liberty 2-4000, Ext. 52335.

<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: center;">UNCLASSIFIED</p>	<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: center;">UNCLASSIFIED</p>
<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: center;">UNCLASSIFIED</p>	<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: center;">UNCLASSIFIED</p>

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: right;">UNCLASSIFIED</p>	<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: right;">UNCLASSIFIED</p>
<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: right;">UNCLASSIFIED</p>	<p style="text-align: center;">UNCLASSIFIED</p> <p>The University of Michigan, Ann Arbor, Michigan. RESEARCH IN AUTOMATA STRUCTURE BEHAVIOR, AND DESIGN, by J. R. Büchi, C. C. Elgot, and J. H. Holland. November 1960. 17 p. (UMRI Project 2794) (Contract DA 36-039-SC-78057) Unclassified report</p> <p>A summary of research on automata design algorithm in formal systems of logic. extensions and alternative (over)</p> <p style="text-align: right;">UNCLASSIFIED</p>

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.*

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNCLASSIFIED

formulations of Kleene automata theory, the role of cycle complexity in nets and finite automata, and systems of theoretical computers with iterative structure and highly parallel action is presented. Abstracts of the seven principal Technical Reports explicating these results are given.

UNCLASSIFIED

UNIVERSITY OF MICHIGAN



3 9015 02651 5281