

UNIVERSAL EMBEDDING SPACES FOR AUTOMATA

John H. Holland

T H E U N I V E R S I T Y O F M I C H I G A N
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Communication Sciences Program

Technical Report

UNIVERSAL EMBEDDING SPACES FOR AUTOMATA

John H. Holland

ORA Project 06114

under contract with:

AF AVIONICS LABORATORY
SYSTEMS ENGINEERING GROUP
AIR FORCE SYSTEMS COMMAND
CONTRACT NO. AF33(615)-1162
WRIGHT-PATTERSON AIR FORCE BASE, OHIO

administered through:

OFFICE OF RESEARCH ADMINISTRATION ANN ARBOR

May 1964

1. INTRODUCTION

In his unfinished manuscript, The Theory of Automata, von Neumann introduces an unusual space obtained by iterating a 29-state automaton to form an infinite 2-dimensional array. The 29-state automaton is so chosen that the resulting space is computation-universal ("logical-universal"): given any computation, there is a pattern of initial states which will cause that computation to be carried out in the space. Although particular examples and classes of computation-universal spaces have been considered since [Burks, 1960; Church, 1957; Holland, 1960-a; Moore, 1962; Myhill, 1962] our knowledge of the role of these spaces in the study of automata and computation remains meager.

In spite of this lack, there can be little doubt of the possibilities offered by universal spaces. Von Neumann's particular usage shows just how far-reaching and unusual these may be: By using his space as a framework within which to represent the construction of automata from simple elements, he was able to give the first careful treatment of self-reproducing automata. In a different vein, computation theory offers several appealing opportunities for the use of universal spaces. For instance, a universal space can serve as a uniform formal context for comparing computation procedures as to complexity and efficiency. The usual approach via Turing machines fixes "work square", "description", and similar conventions at the outset, typically with metalevel statements; thus effects of changes in convention are placed beyond the theory's reach. In contrast, the conventions associated with different computation procedures can be dealt with explicitly in a universal space. This is an important gain since changes in convention drastically affect the procedure employed (the machine required) to carry out a computation.

The present paper seeks to bring the possibilities of universal spaces into sharper focus. It begins (Compositions of Finite Automata) by characterizing the automata which can be built up from an arbitrary finite or countably infinite set of automata. The general idea of representing (or embedding) the structure of one automaton in another is then examined (Embeddings and Computation Procedures). Finally a general class of universal embedding spaces is defined, some important subclasses are distinguished and several properties of the subclasses apropos to studies of computation and construction are investigated (Universal Compositions).

In treating construction, one cannot escape a preliminary commitment to definition of automata in terms of "connected" sets of component automata. A straightforward way to accomplish this is to treat an automaton as a device for transforming sequences into sequences: the two finite functions used to define an automaton, the transition and output functions, can easily be extended to functionals mapping sequences into sequences. (This contrasts with the usual treatment based upon acceptance of finite strings.) "Connection" then amounts to using the output sequence of one automaton as the input sequence of another. A function, called a "composition function" in part 2, can be used to specify which input sequences are to be identified with which output sequences. By restricting the class of composition functions appropriately, it can be assured that:

- (i) the composition of a finite set of automata is an automaton;
- (ii) the composition of a countably infinite set of automata effectively defines a unique output sequence for any component automaton for each choice of free input sequences and assignment of initial states (free input sequence = an input sequence not constrained by the composition function).

Von Neumann's space is an example of a composition on a countably infinite set of copies of a single 29-state automaton. Part 2 gives these ideas a more rigorous formulation. Functionals, of a restricted kind mapping sequences into sequences, become the mathematical objects of study corresponding to the component automata. From the mathematical point of view, then, the composition function induces a set of simultaneous equations on the functionals. These equations correspond to the constraints imposed by the identification of sequences. (In effect, the automaton corresponding to a given composition function is defined by imposing constraining equations on a set of functionals corresponding to a "free" product-automaton.) A composition function is considered admissible only if it satisfies a requirement of "local effectiveness"; this requirement assures a unique solution to the (possibly infinite) set of simultaneous equations. The unique solution defines a new functional. If the set of components is finite, there will exist some finite automaton with transition and output functions extending to this functional. That is, the result of composition in this case is again a finite automaton.

Von Neumann's interest in construction procedures set three important requirements on the space he defined:

- (1) it had to be possible to represent (or embed) an automaton in that space as a connected set of components (that is, the representation had to be "structural" as well as "behavioral" so that one could study processes whereby simpler automata are connected (or composed) to produce more complex automata);
- (2) the space had to be homogeneous (so that the same component could be represented in the same way at any position in the space — an important consideration if construction procedures were not to be dependent upon their location in the space);

(3) given any computation, it had to be possible to represent in the space some automaton capable of carrying out that computation (to avoid a priori restrictions on the procedures which could be represented and, hence, studied).

By defining similar properties for compositions one can characterize a fairly general class of computation-universal spaces. The generalization of property 1 to compositions depends in an essential way upon the notion of embedding — mapping one composition (the object composition) into another (the image composition) with preservation of structure and behavior. Specifically, for a mapping from one composition to another to be an embedding it must satisfy two conditions: the functionals corresponding to a component of the object composition and its image under the mapping must be the same; the images of sequences identified by the composition function of the object composition must be identified by the composition function of the image composition. With this definition homogeneity, property 2, becomes a requirement that, if a composition can be embedded at all, the same embedding procedure must serve mutatis mutandis to embed that composition anywhere in the image composition. Finally, property 3 becomes a requirement that, given any computation, some composition capable of carrying out that computation must be embeddable in the image composition (in what follows this requirement will be restricted to computations which can be carried out by finite automata — a weakening not altering essential points of the discussion). Part 3 of the paper develops these concepts.

The above properties can be combined, as indicated, to define the set, U , of homogeneous, computation-universal compositions. It can easily be shown (part 4) that each member of this set is a composition over a countably infinite set of copies of a single automaton (a single generator) and that the elements

of the composition can be located in a regular fashion in an n -dimensional cartesian coordinate system. A natural subclass, U_d , can be defined by adding as a requirement one further property of the von Neumann space -- a requirement that inputs of the generating automaton affect its outputs only after a non-zero "delay". In other words, the generating automaton is required to be of the Moore type. The subclass U_d includes the natural n -dimensional generalizations of von Neumann's two-dimensional space [von Neumann, unpublished] and counterparts of Church's potentially-infinite automata [Church, 1957]; it excludes "most" of the iterative circuit computers [Holland, 1960-a] although these fall within the class U . A second sub-class, U_c , of U can be defined by strengthening property 3 to a requirement that all finite compositions be embeddable; i.e., elements of U_c must be composition-universal and homogeneous. Certain of the iterative circuit computers belong to the class U_c .

Part 4 investigates and compares U , U_d , and U_c . It is proved there that "most" finite compositions can not be embedded in a composition of class U_d , even if the notion of embedding is considerably weakened. This is so even though every computation can be realized in the space by some embedded composition. The theorem turns on the failure of Kleene's representation theorem [Kleene, 1956] for the embedded compositions. It is a consequence of the theorem for U_d that two compositions which are embeddable in a space of type U_d may be further composed (interconnected) in such a way that the resulting composition is no longer embeddable (assuming a common "input rate" for all images). Stated in another way: Various construction procedures will not be realizable in spaces of type U_d unless "input rates" are permitted to go to zero as the complexity of the embedded compositions increases. A further consequence: Several different computation procedures may employ a

common subroutine, realizable by a single composition, and yet in a space of type U_d the embedded compositions corresponding to the different computations will not have a corresponding common part.

Since there do exist compositions which are not only computation-universal but composition-universal as well, the classes U_d and U_c must be non-empty and disjoint. Hence for a composition to be composition-universal it is necessary that its generating element be of the Mealy type — it must have states for which the input-to-output delay (lag-time) is zero. As a consequence compositions of type U_c , though locally effective, may in some states exhibit arbitrarily long (finite) connected sequences of elements with zero lag-time.

Part 5 provides a closing commentary on definitions and results. Two plans for studying computation and construction procedures are contrasted: One may identify the class of "realistic" universal spaces with the class U_d , restricting the admissible realizations of finite automata (by requiring a less than exponential "fan out" in compositions) so as to retain universality. Alternatively one may identify the class of universal spaces with the class U_c , permitting channels with zero delay. It can be shown that any space of type U_d can be embedded in a space of type U_c ; thus either approach can in fact be implemented via the spaces U_c .

2. COMPOSITIONS OF FINITE AUTOMATA

This study of computation-universal spaces is based upon the possibility of defining an automaton in terms of a "connected" set of component automata. Hence, the first task is to characterize the automata which can be defined by connecting elements of an arbitrary, finite or countably infinite, set of finite automata. A composition function will specify the connections in each case; thus we must characterize the automaton corresponding to each composition function. Since composition is to be defined over arbitrary sets of automata, the sets may include automata of the Mealy type. That is, the output of some component automaton, in some states, may depend upon its current input. (The logical nets of Burks and Wright [Burks and Wright, 1953], which are defined over a set of primitives including "non-delay" switches, are examples of such compositions). In order that the behavior of the resulting compositions be effectively defined it is necessary to restrict the class of composition functions by a requirement of "local effectiveness". Roughly, a composition is locally effective if the state of each component automaton at any time is uniquely determined by the transition functions of automata in some finite "neighborhood" of the component. That is, the state of a component at time $t+1$ can be computed from the internal states and states of free inputs (inputs not constrained by connection), at time t , of a finite, connected subset of elements of the composition — a subset which may in general change with time. In the definitions below most of the complications come from this requirement. However, unless the class of compositions is to be unduly narrowed (excluding thereby "most" logical nets), the requirement cannot be avoided. It will be shown that the behavior of each component of a locally effective composition is uniquely and effectively defined.

Compositions will be defined for any indexed set of elements

$\{a_\alpha \ni \alpha \in A\}$ satisfying the following conditions:

A is a finite or countably infinite ordered set

$$a_\alpha = \langle I_\alpha, S_\alpha, O_\alpha, f_\alpha, u_\alpha \rangle$$

$$I_\alpha = \prod_{i=1}^{m_\alpha} I_{\alpha,i}, \text{ a cartesian product on finite sets } I_{\alpha,i} \quad ;$$

$$S_\alpha, \text{ a finite set} \quad ;$$

$$O_\alpha = \prod_{j=1}^{n_\alpha} O_{\alpha,j}, \text{ a cartesian product on finite sets } O_{\alpha,j} \quad ;$$

$$f_\alpha : I_\alpha \times S_\alpha \rightarrow S_\alpha \quad ;$$

$$u_\alpha : I_\alpha \times S_\alpha \rightarrow O_\alpha \quad .$$

X and Y will designate the (lexicographically ordered) sets of all pairs of indices used to specify sets $I_{\alpha,i}$ and $O_{\alpha,j}$ respectively:

$$X = \{(\alpha, i) \ni \alpha \in A \text{ and } i \leq m_\alpha\}$$

$$Y = \{(\alpha, j) \ni \alpha \in A \text{ and } j \leq n_\alpha\} \quad .$$

Each quintuple a_α with attendant conditions constitutes one of the standard definitions of a finite automaton: the elements of $I_{\alpha,i}$ being interpreted as the states of the i^{th} input of a_α , the elements of S_α as the internal states and the elements of $O_{\alpha,j}$ as the states of the j^{th} output. The quintuple is used to define recursively two families of functionals F_α and U_α (parametrised by S_α) from infinite sequences on I_α (input sequences) to infinite sequences on S_α and O_α respectively (internal state sequences and output sequences):

$$\begin{aligned} F_\alpha &: I_\alpha^N \times S_\alpha \rightarrow S_\alpha^N \\ U_\alpha &: I_\alpha^N \times S_\alpha \rightarrow O_\alpha^N \quad , \quad N = \{0, 1, 2, \dots\} \end{aligned}$$

$$\text{where } I_\alpha^N = \{\underline{I}_\alpha \ni \underline{I}_\alpha : N \rightarrow I_\alpha\}$$

$$S_\alpha^N = \{\underline{S}_\alpha \ni \underline{S}_\alpha : N \rightarrow S_\alpha\}$$

$$O_\alpha^N = \{\underline{O}_\alpha \ni \underline{O}_\alpha : N \rightarrow O_\alpha\}$$

The functions \underline{I}_α , \underline{S}_α , \underline{O}_α are interpreted as infinite sequences. The recursive extension of f_α and u_α to F_α and U_α can be given as follows:

$$F_\alpha(\underline{I}_\alpha(0), s) = s$$

$$F_\alpha(\underline{I}_\alpha(t+1), s) = f_\alpha(\underline{I}_\alpha(t), F_\alpha(\underline{I}_\alpha(t), s))$$

$$U_\alpha(\underline{I}_\alpha(t), s) = u_\alpha(\underline{I}_\alpha(t), F_\alpha(\underline{I}_\alpha(t), s)) \quad s \in S_\alpha, t \in \mathbb{N} \quad .$$

An unrestricted composition, \underline{A} , on $\{a_\alpha\}$ is a quintuple $\langle \underline{I}_A, \underline{S}_A, \underline{O}_A, \underline{f}_A, \underline{u}_A \rangle$

such that:

$$\underline{S}_A = \prod_{\alpha \in A} S_\alpha = \{s : A \rightarrow \prod_{\alpha \in A} S_\alpha \mid \alpha \in A \text{ and } s(\alpha) \in S_\alpha\}$$

$$\underline{I}_A = \prod_{\alpha \in A} \prod_{i=1}^m I_{\alpha, i} = \prod_{x \in X} I_x = \{i : X \rightarrow \prod_{x \in X} I_x \mid x \in X \text{ and } i(x) \in I_x\}$$

$$\underline{O}_A = \prod_{\alpha \in A} \prod_{j=1}^{n_\alpha} O_{\alpha, j} = \prod_{y \in Y} O_y = \{o : Y \rightarrow \prod_{y \in Y} O_y \mid y \in Y \text{ and } o(y) \in O_y\}$$

Under interpretation, $i : X \rightarrow \prod_{x \in X} I_x$ is a particular input state assignment to the inputs indexed by X , and \underline{I}_A is the set of all possible input state assignments (functions); \underline{S}_A and \underline{O}_A are interpreted similarly. The transition function $\underline{f}_A : \underline{I}_A \times \underline{S}_A \rightarrow \underline{S}_A$ is thus a functional, from assignment functions into assignment functions, satisfying the requirement:

$$\underline{f}_A(i, s)(\alpha) = s'(\alpha) = f_\alpha(i(\alpha), s(\alpha))$$

$$\text{where } i \in \underline{I}_A, \quad s, s' \in \underline{S}_A$$

$$i(\alpha) = (i(\alpha, 1), i(\alpha, 2), \dots, i(\alpha, m)) \quad .$$

Similarly $\underline{u}_A : \underline{I}_A \times \underline{S}_A \rightarrow \underline{O}_A$ satisfies

$$\underline{u}_A(i, s)(\alpha) = o(\alpha) = u_\alpha(i(\alpha), s(\alpha)) \quad .$$

Each composition over $\{a_\alpha\}$ will be defined in terms of a composition function γ satisfying the following conditions:

- (i) domain, $Y' \subset Y$
- (ii) range, $X' \subset X$
- (iii) $\gamma : Y' \rightarrow X'$, 1-1 onto

$$(iv) \quad 0_y \subset I_{\gamma(y)} \quad .$$

Under interpretation the function γ determines which inputs of automata in set $\{a_\alpha\}$ are connected to outputs of automata in the set. Property (iii) is a requirement that every output be connected to at most one input — "junctions" or "fan-outs" must be represented by a component automaton with multiple outputs. (Note that $\{a_\alpha\}$ may include "fan-out" elements with no delay: $u_y(i,s) = i$ for $y = (\alpha,1), \dots, (\alpha,n_\alpha)$, where $u_y =_{df} \text{proj}_y u_\alpha(i,s)$). Property (iv) assures the compatibility of the symbols of the output of index y with the symbols of the input of index x — all encoding and decoding procedures must be carried out explicitly by component automata. Each $x \in X'$ in the range of γ selects an input of the unrestricted composition \underline{A} which is to be constrained by the requirement

$$(\forall t) [I_x(t) = 0_{\gamma^{-1}(x)}(t)] \quad .$$

More formally the composition function induces a set of equations of the form

$$U_y(I_\alpha, s) = I_{\gamma(y)} \quad . \quad (*)$$

(For $y = (\alpha, j)$, $U_y(I_\alpha(t), s) =_{df} \text{proj}_y U_\alpha(I_\alpha(t), s)$, an infinite sequence over $0_{\alpha, j}$; $I_{\gamma(y)}$ is defined similarly).

Not all functions γ satisfying conditions (i)-(iv) will yield a consistent set of equations: it may be impossible for the sequences $0_y(t)$, $y \in Y'$, to satisfy both the equations induced by γ and the equations defining the 0_y in terms of the functions u_α . That is, γ may specify the equivalent of a cycle of switches without delay, e.g. a negation element with its output connected to its input. Or, if $\{a_\alpha\}$ is countably infinite, γ may specify a two-way infinite line of switches, none of which involves delay. To assure the consistency of the equations (*), γ must satisfy a further requirement which imposes "local effectiveness" on the equations. This can be accomplished as follows:

A sequence y_1, \dots, y_ℓ such that $y_h \in Y'$, $1 \leq h \leq \ell$, will be called connected if, for $h < \ell$, $\text{proj}_1 \gamma(y_h) = \text{proj}_1 y_{h+1}$ (where $\text{proj}_1 y = \alpha$ if $y = (\alpha, j)$). If $\text{proj}_1 y_1 = \alpha$, $\text{proj}_1 y_\ell = \omega$ and $\ell < \infty$, the connected sequence will be called a sequence from α to ω of length ℓ . A connected sequence may also be infinite, either because an infinite number of distinct elements of Y' are involved or because there exists, say, a sequence y_1, \dots, y_ℓ such that $\text{proj}_1 \gamma(y_\ell) = \text{proj}_1 y_1$ (such as would be generated from elements connected in a cycle by γ , cf. [Holland, 1960]). A connected sequence $\rho = \{y_1, \dots, y_\ell\}$ will be called acyclic if $h \neq h'$ implies $\text{proj}_1 y_h \neq \text{proj}_1 y_{h'}$, $1 \leq h, h' \leq \ell$.

γ will be called consistent with respect to $s \in S_{\underline{A}}$ if, for each $\alpha \in A$, there is an integer $k_{s, \alpha}$ such that every connected sequence to α of length $k_{s, \alpha}$ contains an element $y_h = (\delta, j)$ such that $u_{\delta, j}$ depends only on the state of a_δ , $s(\delta)$, and composition inputs of a_δ , $(\delta, i) \in X-X'$. The requirement that $u_{\delta, j}$ depend only on the state and composition inputs of a_δ can be stated more formally as:

$$(\forall i_1, i_2 \in I_{\underline{A}}) [i_1 | X-X' = i_2 | X-X' \Rightarrow u_{\delta, j}(i_1(\delta), s(\delta)) = u_{\delta, j}(i_2(\delta), s(\delta))]$$

By composition input states (and composition output states) we mean possible state assignments for inputs (and outputs) not constrained by γ :

$$I_{\gamma, \underline{A}} = \{i | X-X' \ni i \in I_{\underline{A}}\}$$

$$O_{\gamma, \underline{A}} = \{o | Y-Y' \ni o \in O_{\underline{A}}\} \quad .$$

Theorem. If γ is consistent with respect to $s \in S_{\underline{A}}$ then each $i' \in I_{\gamma, \underline{A}}$ has a unique extension to an $i \in I_{\underline{A}}$ satisfying equations (*).

Proof outline:

Let $R_{s, \alpha}$ be the set of all finite acyclic sequences to α , $\{y_1, \dots, y_h, \dots, y_\ell\}$, such that $u_{y_1}, y_1 = (\delta, j)$, depends only on $s(\delta)$ and

composition inputs to a_δ , $(\delta, i) \in X-X'$, while each u_{y_h} , $1 > h \geq \ell$, depends upon at least one input indexed by $x \in X'$. (Under interpretation, the first element of each sequence involves a "delay" while the remaining elements are "non-delay" switches, when \underline{A} is a state s .)

If γ is consistent with respect to s then no sequence in $R_{s,\alpha}$ is of length greater than $\ell_{s,\alpha}$ and every connected sequence to α contains some sequence in $R_{s,\alpha}$ as a final sub-sequence. Note that $R_{s,\alpha}$ is finite and can be effectively determined in this case.

Let $B_{s,\alpha} = \{\beta \mid \beta = \text{proj}_1 y \text{ for } y \text{ an element of some sequence of } R_{s,\alpha}\}$. $B_{s,\alpha}$ indexes the set of automata having outputs belonging to sequences in $R_{s,\alpha}$.

For any $i' \in I_{\gamma, \underline{A}}$ and for each x such that $\text{proj}_1 x \in B_{s,\alpha}$, the equations induced by γ determine $i(x)$ uniquely, using just $i'|_{B_{s,\alpha}}$ and $s|_{B_{s,\alpha}}$. Moreover, if $\text{proj}_1 x \in B_{s,\beta}$, $\beta \neq \alpha$, the value for $i(x)$ determined from $B_{s,\beta}$ is the same as that determined from $B_{s,\alpha}$. All of this follows readily upon modifying the algorithm of Theorem X, [Burks and Wright, 1953], so that it applies to the sets $R_{s,\alpha}$.

Since for every $x \in X'$ there exists at least one set $B_{s,\alpha}$ such that $\text{proj}_1 x \in B_{s,\alpha}$, i' can be extended to $i \in I_{\underline{A}}$ as follows:

$$i(x) = \begin{cases} i'(x) & \text{if } x \in X-X' \\ \text{the unique value, } u_{\gamma^{-1}(x)}(i(\text{proj}_1 x), s(\text{proj}_1 x)), & \\ \text{determined by the Burks-Wright algorithm if} & \\ x \in X' & . \end{cases}$$

For γ consistent with respect to s , let $\mu_s(i') \in I_{\underline{A}}$ denote the extension of i' guaranteed by the above theorem.

γ will be called a locally effective composition function (LECF) if the following conditions are satisfied:

- (i) there exists a computable function $s \in S_{\underline{A}}$ for which γ is consistent,
- (ii) if γ is consistent for $s \in S_{\underline{A}}$ then, for every $i' \in I_{\gamma, \underline{A}}$, γ is consistent with respect to $f_{\underline{A}}(\mu_s(i'), s)$.

For γ a LECF define:

$$S_{\gamma, \underline{A}} = \{s \in S_{\underline{A}} \ni \gamma \text{ is consistent w.r.t. } s \text{ and } s \text{ is computable}\}$$

$$f_{\gamma, \underline{A}}(i', s) = f_{\underline{A}}(\mu_s(i'), s)$$

$$u_{\gamma, \underline{A}}(i', s) = u_{\underline{A}}(\mu_s(i'), s) \quad \text{for } i' \in I_{\gamma, \underline{A}}, s \in S_{\gamma, \underline{A}} \quad .$$

For each LECF γ and unrestricted composition \underline{A} the corresponding composition \underline{A}_{γ} , on $\{a_{\alpha}\}$ is the quintuple $\langle I_{\gamma, \underline{A}}, S_{\gamma, \underline{A}}, O_{\gamma, \underline{A}}, f_{\gamma, \underline{A}}, u_{\gamma, \underline{A}} \rangle$. (Where no confusion can arise the subscript \underline{A} will be dropped, i.e., I_{γ} will denote $I_{\gamma, \underline{A}}$, etc.). Thus the set $\{\underline{A}_{\gamma} \ni \underline{A} \text{ is an unrestricted composition and } \gamma \text{ is LECF over } \underline{A}\}$ will be the set of all (finite and countably infinite) compositions (see \underline{A}_{γ} and \underline{B}_{γ_1} of Fig. 1 as examples). The definitions given earlier for finite automata will be used for all compositions; thus $S_{\gamma}^N = \{S_{\gamma} \ni S_{\gamma} : N \rightarrow S_{\gamma}\}$ etc.

Under interpretation, a composition is so defined that there is at least a unit delay in every cycle of elements and in every infinite non-repeating sequence of elements; note that the delays need not appear at the same position each time and that they may, for instance, become less and less "dense" in the space with time.

Theorem. If \underline{A} is over a finite set $\{a_\alpha\}$ then, for any LECF γ over \underline{A} , \underline{A}_γ is a finite automaton.

Theorem. If \underline{A} is over a countably infinite set $\{a_\alpha\}$ then, given any $t \in \mathbb{N}$ and any $\alpha \in A$, $\underline{S}_\gamma(t)|\alpha$ can be calculated from $\underline{S}_\gamma(0)|B(\alpha,t)$ and $\underline{I}_\gamma(0)|B(\alpha,t)$, ..., $\underline{I}_\gamma(t)|B(\alpha,t)$ where B is a computable function and $B(\alpha,t)$ is a finite subset of A containing α .

(Under interpretation, this theorem says that the state of any element a_α of the composition at any time t can be determined from the initial state and composition input sequence up to time t of a finite portion of the composition, indexed by $B(\alpha,t)$ — $B(\alpha,t)$ will in general be an unbounded function of time. Repeated use of the Burks-Wright algorithm enables determination of $B(\alpha,t)$ for any α and t .)

Theorem. The property LECF is decidable for all γ over finite \underline{A} .

(The proof of this theorem follows from a variant of the algorithm for determining admissible states of a finite automaton, cf. [Burks and Wright, 1953]).

Theorem. The property LECF is not decidable for all γ over all \underline{A} .

(The proof of the theorem depends upon the representability of all Turing machines in particular \underline{A} constrained by γ : There exist γ and \underline{A} such that, if a_δ goes into state s^* , γ is not consistent for any s_γ for which $s_\gamma(\delta) = s^*$. Moreover, γ can be so selected that deciding whether a_δ takes on state s^* depends upon a solution of the halting problem. Thus, decidability of LECF for γ would yield a decision procedure for the halting problem).

For some extensive sub-classes over countably infinite sets $\{a_\alpha\}$, such as the class of iterative circuit computers, it can be proved that all associated γ are LECF.

The following definitions will be useful:

$\{a_1, \dots, a_k\}$ will be called a set of generators for \underline{A}_γ if $\{a_\alpha \ni \alpha \in A\}$ consists only of copies of elements of $\{a_1, \dots, a_k\}$.

$\underline{B}_{\gamma'}$ over $\{b_\beta \ni \beta \in B\}$ will be called a subcomposition of \underline{A}_γ over $\{a_\alpha \ni \alpha \in A\}$ if: (i) $\{b_\beta\} \subset \{a_\alpha\}$, (ii) $\gamma' = \gamma|_{\underline{Y}_B}$ where $\underline{Y}_B = \{y \in Y' \ni \text{proj}_1 y \in B \text{ and } \text{proj}_1 \gamma(y) \in B\}$.

3. EMBEDDINGS AND COMPUTATION PROCEDURES

The central objective of an embedding is to mark out, in some (image) composition, a sub-composition which will exhibit the same local behavior as a given (object) composition. For every element in the object composition there must be distinct part (a sub-sub-composition) of the image which defines the same functionals (over images of the free input sequences). Among other things, this amounts to complete preservation of local properties of the information flow. Thus one can study such properties in the embedding. If many compositions can be embedded in a single image composition, various comparisons and interactions can be implemented and studied. Note that only part of the inputs (or outputs) of the image sub-composition may be required for the embedding; also it may be that several inputs (outputs) of the sub-composition will be required to "encode" a single input (or output) of the object composition. This "encoding" causes problems when one tries to reflect the connection scheme of the object composition in the image sub-composition. The definitions are cumbersome largely on this account.

We will consider first a strict notion of embedding which preserves all local behavior on the original time-scale. Afterward this notion will be weakened so that the object composition can be partitioned and only behavior of and connections between elements of the partition need be preserved and this only on a dilated time-scale.

Let \underline{A}_γ be a composition with associated index sets A, X, Y . Let \underline{B}_{γ_1} with associated index sets A_1, X_1, Y_1 be a sub-composition of \underline{C}_{γ_2} . Let ϕ be a mapping from A, X, Y and $S_\gamma, I_\gamma, O_\gamma$ to subsets of $A_1, X_1, Y_1, S_{\gamma_1}, I_{\gamma_1}, O_{\gamma_1}$ respectively (i.e. $\phi(\alpha) \subset A_1, \phi(x) \subset X_1$, etc.). ϕ will be called a strict

embedding of $A_{\underline{\gamma}}$ in $C_{\underline{\gamma}_2}$ (on $B_{\underline{\gamma}_1}$) if the following conditions are satisfied:

(i) Distinct {elements, inputs, outputs} map onto distinct sets of {elements, inputs, outputs}:

$$\alpha \neq \alpha_1 \Rightarrow \phi(\alpha) \cap \phi(\alpha_1) = \Lambda, \quad x \neq x_1 \Rightarrow \phi(x) \cap \phi(x_1) = \Lambda, \quad ,$$

$$\text{and } y \neq y_1 \Rightarrow \phi(y) \cap \phi(y_1) = \Lambda, \text{ where } \Lambda \text{ is the null set.}$$

(ii) Each {input, output, composition ($A_{\underline{\gamma}}$) input, composition ($A_{\underline{\gamma}}$) output} of an element of index α maps onto a set of {inputs, outputs, composition ($B_{\underline{\gamma}_1}$) inputs, composition ($B_{\underline{\gamma}_1}$) outputs} of the subset of elements in $B_{\underline{\gamma}_1}$ indexed by $\phi(\alpha)$:

$$\text{proj}_1 \phi(x) \subset \phi(\text{proj}_1 x), \quad \text{proj}_1 \phi(y) \subset \phi(\text{proj}_1 y)$$

$$x \in X-X' \Rightarrow \phi(x) \subset X_1-X'_1, \quad y \in Y-Y' \Rightarrow \phi(y) \subset Y_1-Y'_1 \quad .$$

(iii) If y is connected to x by γ then all outputs in $\phi(y)$ must be connected by γ_1 to all inputs in $\phi(x)$:

$$\phi(\gamma(y)) = \gamma_1(\phi(y)).$$

(iv) If two {internal, input, output} states of $A_{\underline{\gamma}}$ assign the same state to {element α , input x , output y } then the images of these states must assign the same state to $\{\phi(\alpha), \phi(x), \phi(y)\}$:

$$s(\alpha) = s_1(\alpha) \Rightarrow \phi(s) | \phi(\alpha) = \phi(s_1) | \phi(\alpha)$$

$$i(x) = i_1(x) \Rightarrow \phi(i) | \phi(x) = \phi(i_1) | \phi(x)$$

$$o(y) = o_1(y) \Rightarrow \phi(o) | \phi(y) = \phi(o_1) | \phi(y) \quad .$$

(Note that not all states of $B_{\underline{\gamma}_1}$ need be images of states of $A_{\underline{\gamma}}$).

(v) When $B_{\underline{\gamma}_1}$ is in state $\phi(s)$ the state transition function, $f_{\underline{\gamma}_1}$, depends only on $\phi(s)$ and states assigned to images of composition inputs of $A_{\underline{\gamma}}$:

$$i' | \phi(X-X') = i'_1 | \phi(X-X') \Rightarrow f_{\underline{\gamma}_1}(i', \phi(s)) = f_{\underline{\gamma}_1}(i'_1, \phi(s))$$

$$\text{where } i', i'_1 \in I_{\underline{\gamma}_1} \quad .$$

(vi) The state transition function of the image of α acting upon images of input and internal states of α must yield the image of the successor state produced by the transition function of α , f_α ; similarly for u_α :

$$f_{\phi(\alpha)}(\phi(i) | \phi(\alpha), \phi(s) | \phi(\alpha)) = \phi f_\alpha(i | \alpha, s | \alpha)$$

$$u_{\phi(\alpha)}(\phi(i) | \phi(\alpha), \phi(s) | \phi(\alpha)) = \phi u_\alpha(i | \alpha, s | \alpha)$$

where $f_{\phi(\alpha)}$ is the transition function of the sub-composition indexed by $\phi(\alpha)$ and similarly for $u_{\phi(\alpha)}$.

Conditions (v) and (vi) assure that, for every pair of local transformations (F_α, U_α) in the object composition, there is a sub-composition of the image with transformations $(F_{\phi(\alpha)}, U_{\phi(\alpha)})$ which can be restricted so as to be the same as (F_α, U_α) . Conditions (i)-(iv) assure the proper composition of these transformations in the image. Figure 1 presents an example of a strict embedding.

If ϕ satisfies the following weakened requirements it will be called a weak and b-slow embedding or, briefly, an embedding:

(i)-(vi) All requirements for strict embedding stand except as altered by (vii) and (viii).

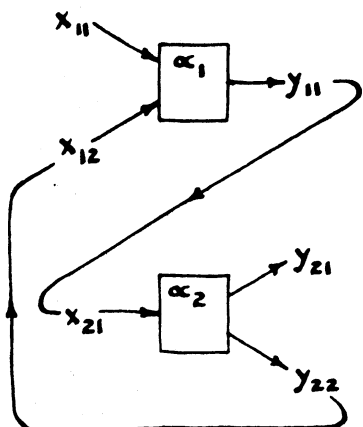
[weak] (vii) Let $\underline{\Lambda}_\gamma$ be partitioned into a set of disjoint sub-compositions. Each such sub-composition may be treated as a single element and be embedded accordingly. Thus γ need only be preserved over connections between sub-compositions. More formally: Let $\underline{\Lambda}_\gamma$ be partitioned by a partitioning, Π , of its index set, Λ . Let $\{\underline{\Lambda}_\gamma | Y_P\}$ represent the set of induced sub-compositions. Then condition (iii) is to apply only to $y \in Y_P - Y'_P$, $P \in \Pi$. All other conditions are modified similarly. (Under interpretation, a weak embedding permits one to ignore all structure and information flow within the sub-compositions $\{\underline{\Lambda}_\gamma | Y_P\}$; only preservation of the associated functionals, $F_{\gamma | Y_P}$ and $U_{\gamma | Y_P}$, is required).

\overline{A}_Y

$$X' = \{x_{12}, x_{21}\}$$

$$Y' = \{y_{11}, y_{22}\}$$

y	y_{11}	y_{22}
$\gamma(y)$	x_{21}	x_{12}

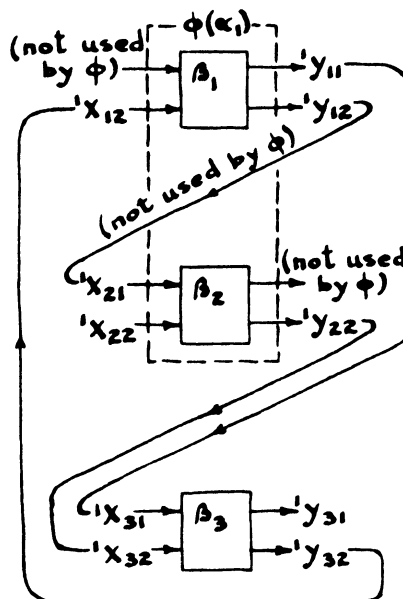


\overline{B}_{γ_1}

$$X'_1 = \{{}^1x_{12}, {}^1x_{21}, {}^1x_{31}, {}^1x_{32}\}$$

$$Y'_1 = \{{}^1y_{11}, {}^1y_{12}, {}^1y_{22}, {}^1y_{32}\}$$

1y	${}^1y_{11}$	${}^1y_{12}$	${}^1y_{22}$	${}^1y_{32}$
$\gamma_1({}^1y)$	${}^1x_{31}$	${}^1x_{21}$	${}^1x_{32}$	${}^1x_{12}$



$$\phi(\alpha_1) = \{\beta_1, \beta_2\}$$

$$\phi(\alpha_2) = \{\beta_3\}$$

$$\phi(y_{11}) = \{{}^1y_{11}, {}^1y_{22}\}$$

$$\phi(y_{21}) = \{{}^1y_{31}\}$$

$$\phi(y_{22}) = \{{}^1y_{32}\}$$

$$\phi(x_{11}) = \{{}^1x_{22}\}$$

$$\phi(x_{12}) = \{{}^1x_{12}\}$$

$$\phi(x_{21}) = \{{}^1x_{31}, {}^1x_{32}\}$$

$$\phi(S_{\alpha_1}) \subset S_{\beta_1} \times S_{\beta_2}, \text{ etc.}$$

[Note: $\text{proj}_1 \phi(y_{11}) = \text{proj}_1 \{{}^1y_{11}, {}^1y_{22}\} = \{\beta_1, \beta_2\} \subset \{\beta_1, \beta_2\} = \phi(\alpha_1) = \phi(\text{proj}_1 y_{11})$

and $\phi(\gamma(y_{11})) = \phi(x_{21}) = \{{}^1x_{31}, {}^1x_{32}\} = \{\gamma_1({}^1y_{11}), \gamma_1({}^1y_{22})\} = \gamma_1(\phi(y_{11}))$

etc., as required.]

Figure 1.

EXAMPLE OF STRICT EMBEDDING

[slow] (viii) A b -slow embedding yields an image with a time scale $t' = bt$, where t is the time index of \underline{A}_γ . That is, if o_γ is the output state of \underline{A}_γ at time t , the output state of the embedded composition will be $\phi(o_\gamma)$ at time bt . To accomplish this, input sequences are "slowed" by the "insertion" of $b-1$ "no-signal" states between signals, cf. [McNaughton, 1962]. Formally, the definition depends upon an extension of f_{γ_1} and u_{γ_1} to strings on I_{γ_1} in the usual way (cf. [Rabin and Scott, 1959]). $I_{\gamma_1}^b$ will denote strings of length b over I_{γ_1} and, if $i^* \in I_{\gamma_1}^{b-1}$, $\phi(i) \cdot i^*$ will denote a string of length b with first (earliest) element $\phi(i)$. The following formal condition for a b -slow embedding weakens requirement (vi):

For all $i \in I_\gamma$ there exists a string $i^* \in I_{\gamma_1}^{b-1}$ such that for all $s \in S_\gamma$ and $y \in Y-Y'$ (with f_{γ_1} and u_{γ_1} extended to $I_{\gamma_1}^b$)

$$f_{\gamma_1}(\phi(i) \cdot i^*, \phi(s)) \mid \phi(\alpha) = \phi(f_\gamma(i, s)(\alpha))$$

$$u_{\gamma_1}(\phi(i) \cdot i^*, \phi(s)) \mid \phi(y) = \phi(u_\gamma(i, s)(y))$$

Thus the strings $\phi(i) \cdot i^*$ of length b become the images of the input states i . The conditions assure that, if $\underline{I}_{\gamma_1} = \phi(\underline{I}_\gamma)$, then, for all t , $\underline{S}_{\gamma_1}(bt) = \phi(\underline{S}_\gamma(t))$ and $\underline{O}_{\gamma_1}(bt) = \phi(\underline{O}_\gamma(t))$. (Under interpretation

the signal-processing rate of the image is "slowed" by a factor b).

The reader can develop a considerable understanding of slow embeddings by a close reading of McNaughton's discussion of "slow automata" [McNaughton, 1962 and draft].

The following additional definitions will be useful:

An isomorphic embedding ϕ of $\underline{A}_{\underline{\gamma}}$ on $\underline{B}_{\underline{\gamma}_1}$ is a strict embedding such that:

$$\begin{aligned} \alpha &\approx \phi(\alpha) & x &\approx \phi(x) & y &\approx \phi(y) \\ S_{\alpha} &\approx S_{\phi(\alpha)} & I_x &\approx I_{\phi(x)} & O_y &\approx O_{\phi(y)} \end{aligned}$$

Two embeddings, ϕ_1 and ϕ_2 , of $\underline{A}_{\underline{\gamma}}$ in $\underline{C}_{\underline{\gamma}_2}$ will be called identically oriented if:

- (i) there is an isomorphic embedding, θ , of the sub-composition indexed by $\phi_1(A)$ on the sub-composition indexed by $\phi_2(A)$ such that $\theta\phi_1 = \phi_2$;
- (ii) for all $\alpha, \alpha' \in A$ and for any connected sequence ρ {from $\phi_1(\alpha)$ to $\phi_1(\alpha')$, from $\phi_1(\alpha)$ to $\phi_2(\alpha)$ } there exists a connected sequence ρ' {from $\phi_2(\alpha)$ to $\phi_2(\alpha')$, from $\phi_1(\alpha')$ to $\phi_2(\alpha')$ } such that $\text{proj}_2 \rho = \text{proj}_2 \rho'$ (where $\text{proj}_2 \rho = (i_1, \dots, i_h, \dots, i_\ell)$ just in case $y_h = (\beta_h, i_h)$, $1 \leq h \leq \ell$).

The characterization of homogeneous compositions (in the next section) will be based upon this definition of identical orientation.

Prior to defining computation-universality for compositions we also need a relevant definition of computation. The definition which follows is related to that of Burks [Burks, 1960]:

Let $\Sigma_j = \{\sigma : N \rightarrow N_j \mid N_j = \{0, 1, \dots, j\}\}$. Let $\Gamma : \Sigma_i \rightarrow \Sigma_j$ be a functional from sequences into sequences. We wish to define a finite computation procedure for Γ (of course, only some Γ will admit of such procedures).

Augment a composition $\underline{A}_{\underline{\gamma}}$ by two mappings:

$$\begin{aligned} \lambda_1 : N_i \cup \{\Omega\} &\rightarrow I'_{\underline{\gamma}} & \text{1-1 onto,} & & I'_{\underline{\gamma}} &\subset I_{\underline{\gamma}} \\ \lambda_2 : O'_{\underline{\gamma}} &\rightarrow N_j \cup \{\Omega\} & \text{1-1 onto,} & & O'_{\underline{\gamma}} &\subset O_{\underline{\gamma}} \end{aligned}$$

The symbol ' Ω ' under interpretation determines the "non-signal" states, The mappings

λ_1 and λ_2 are extended to infinite sequences $\sigma \in \Sigma_i$ and $\underline{O}_\gamma \in O_\gamma^N$ (treating sequences as strings and allowing "signals" to be separated by "no signal" sequences) as follows:

$$\begin{aligned}\lambda_1^1(n) &= \lambda_1(n), & \lambda_1^b(n) &= \lambda_1^{b-1}(n) \cdot \lambda_1(\Omega), & n &\in N_i \cup \{\Omega\} \\ \lambda_1(\sigma, b, \tau, s) &= ([\lambda_1^\tau(\Omega) \cdot \lambda_1^b(\sigma(1)) \cdot \lambda_1^b(\sigma(2)) \cdot \dots], s), & s &\in S_\gamma \\ \lambda_2(\underline{O}_\gamma) &= \lambda_2(\underline{O}_\gamma(t_1)) \cdot \lambda_2(\underline{O}_\gamma(t_2)) \cdot \dots\end{aligned}$$

where $\lambda_2(\underline{O}_\gamma(t)) \neq \Omega$ just in case $t = t_j$.

$\langle \underline{A}_\gamma, \lambda_1, \lambda_2, s \rangle$ will be called a b-uniform computation procedure for Γ and Γ will be called (b-) uniformly computable if

$$\begin{aligned}(\forall \sigma \in \Sigma_i, \tau \in N) (\exists \tau' \in N) [\lambda_{2 \gamma 1} \lambda(\sigma, b, \tau, s) = \Gamma(\sigma)] \\ \text{and } [\lambda_{\gamma 1} \lambda(\sigma, b, \tau, s)(t) \neq \Omega \text{ just in case } t = \tau' + jb]\end{aligned}$$

That is, at the rate of one input (output) "signal" every b time-steps, and independent of elapsed time τ before the first signal, \underline{A}_γ must transform signals corresponding to successive elements of σ into signals corresponding to successive elements of $\Gamma(\sigma)$.

The set of finitely computable Γ could be broadened considerably (and naturally) by using $\lambda_1^* : N_i^d \rightarrow I_\gamma^b$ and $\lambda_2^* : O_\gamma^b \rightarrow N_j^d$ in place of λ_1 and λ_2 . If, under quite general conventions, some finite \underline{A}_γ can compute Γ then one can show that \underline{A}_γ can be composed with an acyclic (encoding) composition $\underline{E}_{\gamma 1}$ so that the resultant composition computes Γ via λ_1^* and λ_2^* . Thus the extended definition would serve well for the study of finitely computable Γ . However, this line will be avoided here because it is more complicated and because the theorems and corollaries of part 4 hold for any class of finitely computable Γ containing the uniformly computable Γ as a subclass.

Theorem: If Γ is finitely computable in the sense of [Burks, 1960] then Γ is uniformly computable.

Theorem. Corresponding to each regular event over a given alphabet there is a distinct uniformly computable Γ .

Note that the elements of Σ_j can be interpreted as representations of fractional real numbers. With appropriate restrictions various non-trivial uniformly computable Γ can be interpreted as continuous real functions.

Lemma. Let $\Gamma : \Sigma_i \rightarrow \Sigma_j$ and $\Gamma' : \Sigma_j \rightarrow \Sigma_k$ be arbitrary b -uniformly computable functionals except that the range of Γ is the domain of Γ' . Let $\langle \underline{A}_\Gamma, \lambda_1, \lambda_2, s \rangle$ and $\langle \underline{A}'_{\Gamma'}, \lambda'_1, \lambda'_2, s' \rangle$ be arbitrary b -uniform computations for Γ and Γ' respectively. Then there exists an acyclic composition \underline{E}_{Γ_1} such that:

- (i) the composition outputs of \underline{A}_Γ can be identified in 1-1 fashion with the composition inputs of \underline{E}_{Γ_1} and the composition outputs of \underline{E}_{Γ_1} can be identified in 1-1 fashion with the composition inputs of $\underline{A}'_{\Gamma'}$, to form a new composition \underline{C}_{Γ_2} involving only $\underline{A}_\Gamma, \underline{A}'_{\Gamma'}$, and \underline{E}_{Γ_1} ,
- (ii) the output functional of \underline{C}_{Γ_2} satisfies the equation

$$U_{\Gamma_2} = U_{\Gamma'} U_{\Gamma_1} U_{\Gamma} ,$$

- (iii) $\langle \underline{C}_{\Gamma_2}, \lambda_1, \lambda'_2, s'' \rangle$, for an appropriate $s'' \in S_{\Gamma_2}$, is a b -uniform computation procedure for $\Gamma'\Gamma$ since

$$(\forall \sigma \in \Sigma_i, \tau \in \mathbb{N}) [\lambda'_2 U_{\Gamma'} U_{\Gamma_1} U_{\Gamma} \lambda_1 (\sigma, b, \tau, s'') = \Gamma'\Gamma(\sigma)] .$$

(Under interpretation the output of \underline{A}_Γ can be connected to the input of $\underline{A}'_{\Gamma'}$, via the combinatorial recoding \underline{E}_{Γ_1} so that the resulting composition computes the composition of the functionals Γ and Γ' .)

Theorem. Let $b(\underline{A}_\Gamma, \Gamma) = \min_b \{b \mid \underline{A}_\Gamma \text{ can } b\text{-uniformly compute } \Gamma\}$, undefined if \underline{A}_Γ cannot b -uniformly compute Γ . Given Γ, Γ' , and \underline{C}_{Γ_2} as in the previous lemma, \underline{C}_{Γ_2} can at best b_0 -uniformly compute $\Gamma'\Gamma$ where $b_0 = \max\{b(\underline{A}_\Gamma, \Gamma), b(\underline{A}'_{\Gamma'}, \Gamma')\}$. (The "speed" of the faster computation procedure is wasted).

Theorem. If \underline{A}_Γ can b -uniformly compute Γ and if $\phi(\underline{A}_\Gamma)$ is a b' -slow embedding of \underline{A}_Γ then $\phi(\underline{A}_\Gamma)$ can $b'b$ -uniformly compute Γ .

4. UNIVERSAL COMPOSITIONS

Using the definitions thus far set down we can at last attain a definition, fairly general and fairly concise, of the principal objects of study.

A composition \underline{V}_V of index V will be called universal and homogeneous for (uniformly) computable Γ or computation-universal if:

(i) for each (uniformly) computable Γ there is an embedding ϕ , into \underline{V}_V , of a composition \underline{A}_Γ capable of (uniformly) computing Γ (It is assumed that the class of computable Γ , however defined, includes the class of uniformly computable Γ .);

(ii) if ϕ_1 embeds \underline{A}_Γ of index A in \underline{V}_V then, given arbitrary $\alpha \in A$, $\xi \in \phi(\alpha)$ and $\xi' \in V$, there exists ϕ_2 embedding \underline{A}_Γ in \underline{V}_V with identical orientation so that

$$\theta(\xi) = \xi'$$

(where θ is the isomorphic embedding specified by the definition of identical orientation).

Condition (ii) assures that, if a composition can be embedded at all, it can be embedded anywhere in the space with identical orientation. Thus, in a sense made precise in the next theorem, all "translations" of the original image are also images. As a result, properties of the image such as its connection scheme can be made independent of the "location" of the image. If the universal space is to be used to study growing automata and construction procedures this requirement is critical.

A composition \underline{W}_W of index W will be called universal and homogeneous for finite compositions or, briefly, composition-universal if:

- (i) given any finite composition \underline{A}_γ there exists an embedding ϕ of \underline{A}_γ in \underline{W}_ζ ;
- (ii) as for computation-universal.

The work of von Neumann [von Neumann, unpublished] establishes the existence of computation-universal compositions. It can also be shown that there exist composition-universal compositions (the class of iterative circuit computers [Holland, 1960] is an effectively defined class of compositions containing a subset of composition-universal compositions). Note that the class of composition-universal compositions, U_c , is a subclass of the class of computation-universal compositions, U .

Theorem. The following conditions are necessary for a composition \underline{V}_ν to be computation-universal:

- (i) the composition must be generated by a single element,
- (ii) it must be over a countably infinite set of elements,
- (iii) strings of output indices are commutative, i.e. if $\sigma = \text{proj}_2 \rho$ where ρ is a connected sequence from α to β , then for each permutation σ' of σ there exists a connected sequence ρ' from α to β such that $\sigma' = \text{proj}_2 \rho'$.

Proof outline:

For (i), let a_α and a_β be any two elements of \underline{V}_ν and consider the one-element composition \underline{A}_γ which is simply a copy of a_α . Obviously there exists ϕ_1 embedding \underline{A}_γ at α in \underline{V}_ν . Hence, by the second condition in the definition of computation-universal, there must be an embedding ϕ_2 of \underline{A}_γ at β and an isomorphic embedding θ of $\phi_1(\underline{A}_\gamma) = a_\alpha$ at β such that $\theta\phi_1 = \phi_2$. It follows immediately that the quintuples $\langle I_\beta, S_\beta, O_\beta, f_\beta, u_\beta \rangle$ and $\langle I_\alpha, S_\alpha, O_\alpha, f_\alpha, u_\alpha \rangle$ must be identical.

(ii) follows at once from (i) and the existence, for arbitrary k , of Γ computable only by compositions \underline{A}_γ wherein the cardinality of S_γ exceeds k .

(iii) follows from the common cardinality of all input (output) index sets (since \underline{V}_v is generated by a single element) and the following argument: Let $\rho = (y_1, y_2)$ be any two-element connected sequence in \underline{V}_v with, say, $y_1 = (\alpha, i)$, $y_2 = (\beta, j)$, and $\text{proj}_1 v(y_2) = \delta$. Let θ be an identically oriented isomorphic embedding of the sub-composition on elements α, β with $\theta(\beta) = \delta$. But then y_2 constitutes a one-element connected sequence from β to δ whence by identical orientation there must be a connected sequence ρ_1 from α to $\theta(\alpha)$ such that $\text{proj}_2 \rho_1 = \text{proj}_2 y_2 = j$. Moreover, since y_1 constitutes a connected sequence from α to β , there must be a connected sequence ρ_2 from $\theta(\alpha)$ to $\theta(\beta)$ such that $\text{proj}_2 \rho_2 = \text{proj}_2 y_1 = i$. The sequence $\rho' = (\rho_2, \rho_1)$ is connected, from α to $\theta(\beta) = \delta$, and $\text{proj}_2 \rho' = (j, i)$. Since $\text{proj}_2 \rho = (i, j)$, with i, j arbitrarily chosen, this proves commutativity of indices for arbitrary pairs. From this, commutativity for arbitrary strings of indices can be established in the usual way.

————— * —————

It is a consequence of this theorem that we can "co-ordinatise" the computation-universal spaces in terms of a subset $\{n_1, \dots, n_k\}$, $n_k \cong n_\alpha$, of the output indices of the generator: Define a non-oriented connected sequence from α to β to be the same as a connected sequence except that, for each y_h , either $\text{proj}_1(y_{h+1}) = \text{proj}_1 v(y_h)$ or $\text{proj}_1(y_h) = \text{proj}_1 v(y_{h+1})$ is permitted. Count an index occurrence as -1 in the latter case. Choosing some α as origin, the position of any α' is given by a k -tuple having its j^{th} component equal to the sum of the (signed) occurrences of index n_j in any non-oriented connected sequence from α to α' . The above theorem assures the

uniqueness of this number; the resulting co-ordinatization is the usual k -dimensional (integral) cartesian grid.

The theorem also enables us to define simply one further class of compositions:

$U_d = \{ \underline{V}_\gamma \rightarrow \underline{V}_\gamma \text{ is computation-universal with a Moore automaton as generating element} \}$.

Elements of U_d are the most natural generalizations of von Neumann's "logical-universal" space because each exhibits the "propagation-delay" or non-zero lag time which plays an important part in von Neumann's development. Referring to the definition of slow embedding, lag time can be more formally defined:

a composition \underline{A}_γ has lag-time b if (extending u_γ to strings of length $b+1$):

$$(\exists i, i' \in I_\gamma) (\forall s \in S_\gamma) (\forall i^*, i^{**} \in I_\gamma^b) [u_\gamma(i^*, s) = u_\gamma(i^{**}, s) \text{ and } u_\gamma(i \cdot i^*, s) \neq u_\gamma(i' \cdot i^*, s)] .$$

If $\underline{V}_\gamma \in U_d$ has a generator with lag-time b and if the shortest connected sequence from α to β in \underline{V}_γ has length ℓ , it will take at least $b\ell$ units of time for the output of element a_β to be affected by an input to a_α .

The central result of this section establishes the set U_d as disjoint from the set U_c of composition-universal compositions; i.e. elements of U_d are not composition-universal. It follows at once that the generating element of any composition of type U_c must be a Mealy type automaton. The result actually shows that, given any finitely computable Γ , "almost none" of the compositions capable of computing Γ can be embedded in a composition of type U_d . This has several consequences, stated as corollaries and discussed thereafter, for the study of computation (and construction) procedures in computation-universal spaces.

In the following discussion: \underline{V}_γ will be a composition of type U_d on a generator g with n outputs and lag-time $\tau_g \geq 1$; G will be a finite set of

automata, arbitrary except that it contains at least one element with two or more input-dependent outputs (every set of generators complete for all finitely computable Γ must contain at least one such element); τ_G will be the minimum lag-time for elements of G having two or more outputs; $[\underline{A}_\gamma]_{\beta, \ell}$ will be the (unique) sub-composition of \underline{A}_γ consisting of all elements a_α of \underline{A}_γ such that, for some connected sequence ρ from β of length ℓ , α occurs in $\text{proj}_1 \rho$ (i.e. $[\underline{A}_\gamma]_{\beta, \ell}$ is the sub-composition consisting of all elements belonging to connected sequences of length ℓ from β).

Theorem. Given any \underline{V}_ν , G , and $b \in \mathbb{N}$ there exists a composition \underline{A}_γ generated by G which cannot be embedded in \underline{V}_ν . (In fact for each \underline{V}_ν , G , and b there exists a constant c and a function $e(\ell)$ such that the theorem holds for any \underline{A}_γ containing a sub-composition $[\underline{A}_\gamma]_{\beta, \ell}$, $\ell \geq c$, with at least $e(\ell)$ elements; the proof shows that $e(\ell)$ and c are such that "almost all" \underline{A}_γ over G satisfy the condition).

Proof outline:

Given \underline{V}_ν and G there will be a positive integer k such that no composition \underline{B}_γ over G consisting of more than k elements can be weakly embedded in the generator g of \underline{V}_ν .

Because of commutativity of strings of output indices in \underline{V}_ν , fewer than $(2\ell)^n$ distinct elements can belong to any sub-composition $[\underline{V}_\nu]_{\beta, \ell}$ of \underline{V}_ν (where n is the number of outputs of g).

But then any composition \underline{A}_γ over G containing more than $k(2\ell)^n$ elements will have an image in \underline{V}_ν under weak embedding with at least one acyclic connected sequence of length $> \ell$.

There are compositions \underline{A}_γ over G consisting of $2^{\ell+1}-1$ elements and having no acyclic sequence of length $> \ell$.

Since there exists c_0 such that $k(2\ell)^n < 2^{\ell+1}-1$ for $\ell \geq c_0$, there are compositions over G which can only be embedded in \underline{V}_v if some acyclic sequence in the image has length $\ell + \ell_1 > \ell$.

By simply increasing c_0 we can show, in fact, that there exists c for which, when $\ell \geq c$

$$(\ell + \ell_1) \tau_g \geq \max(\ell \tau_G, b) \quad ;$$

thus for any \underline{A}_γ over G consisting of more than $k(2\ell)^n$ elements, with $\ell \geq c$, the lag-time $\tau_{\phi(\underline{A}_\gamma)}$ of the image under an embedding ϕ must exceed both b and the lag-time $\tau_{\underline{A}_\gamma}$ of \underline{A}_γ :

$$\tau_{\phi(\underline{A}_\gamma)} > \max(\tau_{\underline{A}_\gamma}, b) \quad .$$

Therefore, if \underline{A}_γ as specified is to be embedded, the embedding ϕ must be slow since other types of embedding (weak but not slow) preserve lag-time.

Let \underline{C}_{γ_2} contain \underline{A}_γ as a sub-composition in such a way that some composition outputs of \underline{C}_{γ_2} depend upon composition outputs of \underline{A}_γ . Since the lag-time of $\phi(\underline{A}_\gamma)$ will be $\tau_{\phi(\underline{A}_\gamma)} > \tau_{\underline{A}_\gamma}$, the overall lag-time of composition outputs of $\phi(\underline{C}_{\gamma_2})$ must exceed that of \underline{C}_{γ_2} . Otherwise the slow embedding requirement that composition output values occur synchronously will not be satisfied. (Under interpretation: since signals through $\phi(\underline{A}_\gamma)$ are unduly delayed, the proper phasing of signals at the output can only be restored by delaying the other signals accordingly.) The result will be an overall increase in the lag-time τ of $\phi(\underline{C}_{\gamma_2})$ to at least $\tau_{\phi(\underline{A}_\gamma)} > b$.

Consider the image $\phi(\underline{C}_{\gamma_2})$ of \underline{C}_{γ_2} in \underline{V}_v under a b -slow embedding ϕ . For the moment assume $2b \geq \tau$. Because $\tau > b$, $\phi(\underline{S}_{\gamma_2}(t))$ can at best be determined in $\phi(\underline{C}_{\gamma_2})$ at time $(b-1)t + \tau > bt$, even if $(\underline{S}_{\gamma_2}(t-1))$ is available at time $(b-1)t$. In fact, in order that $\phi(\underline{S}_{\gamma_2}(t-1))$ always be determined by time

bt, for arbitrary t, it is necessary that $\phi(\underline{S}_{\gamma_2}(t+1))$ be determined directly from $\phi(\underline{S}_{\gamma_2}(t-1))$ rather than from $\phi(\underline{S}_{\gamma_2}(t))$. This is only possible if the transition function of the image, in effect, has images of $\underline{S}_{\gamma_2}(t-1)$, $\underline{I}_{\gamma_2}(t-1)$ and $\underline{I}_{\gamma_2}(t)$ as arguments at time bt — a sufficient (and for particular \underline{C}_{γ_2} , necessary) condition for determination of $\phi(\underline{S}_{\gamma_2}(t+1))$. $\phi(\underline{I}_{\gamma_2}(t))$ will be available at time bt, but $\underline{I}_{\gamma_2}(t-1)$ will be available only if the set of states of the image is the image of the set $\underline{I}_{\gamma_2} \times \underline{S}_{\gamma_2}$. (Under interpretation: new storage elements would have to be added to the embedded network to "hold" the image of $\underline{I}_{\gamma_2}(t-1)$). This violates requirements (iv) and (vi) of the definition of embedding. Yet all of this is necessary to preserve the function U_{γ_2} under ϕ (by meeting the requirement that the image of $\underline{O}_{\gamma_2}(t)$ always occur at time bt in \underline{V}_v under a b-slow embedding). Hence the assumption that \underline{C}_{γ_2} , as specified, can be b-slow embedded in \underline{V}_v leads to a contradiction.

A similar argument requiring $\underline{S}_{\gamma_2}(t-k_1)$, $\underline{I}_{\gamma_2}(t-k_1)$, $\underline{I}_{\gamma_2}(t-k_1+1)$, ..., $\underline{I}_{\gamma_2}(t)$ applies when $(k+1)b \geq \tau > kb$.

———— * ————

Note that "almost all" compositions on G as specified will include a sub-composition $[\underline{A}_{\gamma}]_{\beta, \ell}$, $\ell \geq c$, containing more than $k(2\ell)^n$ elements.

Corollary. Given any finite composition \underline{A}_{γ} and any b-slow embedding ϕ of \underline{A}_{γ} in space \underline{V}_v of type U_d , there exists a composition \underline{A}'_{γ} , which contains \underline{A}_{γ} as a sub-composition and which cannot be b-slow embedded in \underline{V}_v .

In fact, given any $b \in \mathbb{N}$, "almost all" compositions containing \underline{A}_{γ} cannot be b-slow embedded in \underline{V}_v . Thus most schemes for composing \underline{A}_v with other compositions cannot be represented in \underline{V}_v using b-slow embeddings, even when \underline{A}_v itself is embeddable.

Corollary. Given $\underline{V}_v \in U_d$, any $b \in \mathbb{N}$, and any set of generators G as specified above, there exists an infinite sequence of functionals $\Gamma_1, \Gamma_2, \dots, \Gamma_j, \dots$ such that:

- (i) each Γ_j is b -uniformly computable by a composition $[\underline{A}_j]_{\gamma_j}$ over G ,
- (ii) $[\underline{A}_j]_{\gamma_j}$ can at best be b_j -slow embedded in \underline{V}_v where $b_j > b_{j-1}$, $b_j \in \mathbb{N}$.

Note that $\lim_j b_j = \infty$ so that the computation rates of the images of the $[\underline{A}_j]_{\gamma_j}$ in \underline{V}_v must approach zero. If \underline{A}_γ can b -uniformly compute Γ then a composition $[\underline{C}_j]_{\gamma_j}$, containing \underline{A}_γ and $[\underline{A}_j]_{\gamma_j}$ as subcompositions, can b -uniformly compute $\Gamma'_j = \Gamma_j \Gamma$. However there is no $b' \in \mathbb{N}$ such that all $[\underline{C}_j]_{\gamma_j}$ can be b' -slow embedded in \underline{V}_v . Hence, although \underline{A}_γ can serve as a common "sub-routine" for computing all Γ'_j , there need be no corresponding common sub-routine for computations of the Γ'_j in \underline{V}_v (assuming a minimal signal rate $1/b'$ specified).

The restrictions just noted do not hold for the composition-universal spaces U_c . In fact there exist composition-universal spaces such that any finite composition can be strictly embedded in the space. The spaces U_c are compositions and hence satisfy local effectiveness conditions; however, each such space is generated by a Mealy-type automaton, and may exhibit arbitrarily long (finite) connected sequences of elements with zero lag-time (cf. the similar situation for logical nets).

Theorem. Given any composition $\underline{V}_v \in U_d$ there exists a composition $\underline{C}_\gamma \in U_c$ in which \underline{V}_v can be strictly embedded.

Proof outline:

As indicated earlier all compositions in U can be given Cartesian coordinates; one can then determine a unique dimension for each. \underline{V}_v has a single finite automaton as generator and that generator can be strictly embedded

in a finite n -cube of some space $\underline{C}_\gamma \in U_c$ of dimension n . If the dimension of \underline{V}_ν is less than or equal to n , a set of image n -cubes can be arrayed in \underline{C}_γ with the same geometry as the elements of \underline{V}_ν .

5. COMMENTARY

The central definitions of this paper are structural -- involving no statements quantified over time. Thus, the compositions are defined via a composition function which specifies the graph of connections between component automata, the behavior of the resulting structure being defined by a set of simultaneous equations induced by the composition function. An embedding selects a sub-composition of an image composition and constrains certain of its state components. These constraints are invariant (hereditary) under a natural restriction of the transition function of the image composition: a restriction wherein input states to the sub-composition are restricted to images of the input states of the object composition. Only when the inputs to the image depart from this condition will the embedding cease to reflect the behavior of the object composition.

The full set of finite compositions may for some purposes be too broad, containing "unrealistic" elements. A natural way to achieve a "realistic" subset is to admit only compositions over a selected set of "realistic" generators. For example, a generator may be designated "realistic" only if it has non-zero lag-time (the Moore type automata). The results of part 4 are intrinsic in this respect: No matter how one limits the set of "realistic" elements by selection of generators, almost none of the compositions over the selected generators will b' -slow embeddable, for any $b' \in \mathbb{N}$, in any $\underline{V}_v \in U_d$. Thus, given any $b' \in \mathbb{N}$, almost none of the b -uniform computation procedures over the generators can be embedded in \underline{V}_v as $b'b$ -uniform computation procedures. Instead of a computation rate $1/b$, almost all images will have

a rate less than ϵ , for any $\epsilon > 0$ — an undesirable consequence if the space is to be used to study computation procedures. Two alternatives for avoiding this consequence present themselves:

- (i) One can accept the spaces U_d as the only "realistic" universal spaces and avoid the above consequence by further narrowing the set of admissible representations of finite automata. This can not be achieved by selecting a set of "realistic" primitives; it can only be done by restricting the connection procedures. (The heart of the difficulty lies in the unrestricted cascading of fanout elements.) Such a restriction would considerably modify current notions of representation and several theorems, such as the Kleene representation theorem for regular events (Theorem 3 [Kleene, 1956]), would be lost.
- (ii) One can use the spaces U_c with their concomitant of connected sequences of elements exhibiting negligible delay. This introduces no mathematical difficulty because the spaces U_c are compositions and hence satisfy strong local effectiveness conditions; nevertheless one may question how "realistic" such spaces are in allowing propagation of signals with negligible delay over arbitrarily long channels.

The second alternative seems preferable to me: Criteria which argue for alternative (i) can be applied, via the last theorem of part 4, to restrict the ^{function}transformation of spaces of type U_c . Thus such criteria can both be implemented and studied under the second alternative. At the same time, there are useful restrictions of U_c not readily translatable to U_d .

In this context the study of construction procedures, or growing automata (leading to studies of self-reproduction, adaptation, etc.) becomes the study of sequences of compositions. Such sequences become state sequences in a universal composition — an embedding must transform structural features of

the object composition into properties of the state pattern of the universal composition. The fixed connection scheme of the universal composition can only supply an invariant framework. Note that the state sequence corresponding to a sequence of compositions must involve departures from the constraints imposed by embedding. That is, the image of one composition can be altered to that of another only by departing from the conditions set by the embedding. A restriction of the transition function of the universal space will still be involved; it will be similar to, but weaker than, that required for embeddings. The restriction must be such that any state trajectory under this restricted function corresponds to a sequence of compositions. Note that a construction procedure involving structures that cannot be embedded will not be representable. More than this, given a universal space, there will be construction procedures, involving only embeddable structures, which still fail of representation. That is, some recursive enumerations of embeddable structures will not be representable through a restriction of the transition function. This is not necessarily undesirable: If it is assumed that the "size" of given composition in a construction sequence is a function of the "size" of its predecessor, then most recursive enumerations of compositions will not be construction sequences. Questions related to characterization of the construction procedures representable in various universal spaces remain open.

ACKNOWLEDGEMENTS

I would like to thank Robert McNaughton for sending me an early draft of his extensive work on badly timed elements (a portion of which appeared in 1962 [McNaughton, 1962]); the work he has done and the work reported here are complementary in several parts. I would like to thank James Thatcher for many helpful comments and suggestions based on a careful reading of an early draft of the manuscript; in several places the notation derives directly from his comments. The research reported here was in part supported by U.S. Air Force contract AF 33(615)-1162.

REFERENCES

- Burks, A. W., (1960): Computation, behavior and structure in fixed and growing automata. Self-Organizing Systems. Marshall Yovits and Scott Cameron, Editors. New York, Pergamon Press (pp. 282-311).
- Burks, A. W., and Wright, J. B., (1953); Theory of logical nets. Proceedings of the Institute for Radio Engineers, 41, 1357-1365.
- Church, A., (1957): Application of recursive arithmetic to the problem of circuit synthesis. Summaries, Summer Institute for Symbolic Logic. Cornell University, Institute for Defense Analysis (pp. 3-50).
- Hartmanis, J., (1961); On the state assignment problem for sequential machines. Institute for Radio Engineers, Transactions on Electronic Computers EC-10, 3, 157-165.
- Holland, J. H., (1960); Cycles in logical nets. Journal of the Franklin Institute, 270, 3, 202-226.
- Holland, J. H., (1960-a): Iterative circuit computers. Proceedings of the Western Joint Computer Conference, San Francisco (pp. 259-265).
- Kleene, S. C., (1956): Representation of events in nerve nets and finite automata. Automata Studies. C. E. Shannon and J. McCarthy, Editors. Princeton, New Jersey. Princeton University Press (pp. 3-41).
- McNaughton, R., (1962): On nets made up of badly timed elements, I. Summer Conference Notes on Parallel Computers, Automata and Adaptive Systems. Ann Arbor. The University of Michigan.
- McNaughton, R., (draft): On nets made up of badly timed elements. Continuation of On nets made up of badly timed elements, I.
- Moore, E. F., (1962): Machine models of self-reproduction. Mathematical problems in the biological sciences. Providence. American Mathematical Society (pp. 17-33).

- Myhill, J., (1962): Self-reproducing automata. Summer Conference Notes on Parallel Computers, Automata and Adaptive Systems, Ann Arbor. The University of Michigan.
- Rabin, M. O., and Scott, D., (1959); Finite automata and their decision problems. IBM Journal of Research and Development, 3, 114-125.
- von Neumann, J., (unpublished): The theory of automata, construction, reproduction homogeneity.

