THE UNIVERSITY OF MICHIGAN
COLLEGE OF LITERATURE, SCIENCE, AND THE ARTS
Computer and Communication Sciences Department

COMPUTATION, SELF-DESCRIPTION, CONSTRUCTION, AND REPRODUCTION IN SOME ARTIFICIAL MOLECULAR MACHINE SYSTEMS

Richard A. Laing

February 1975

Technical Report No. 167

with assistance from

National Science Foundation
Grant No. DCR71-01997
Washington, D.C.

Abstract

In Part I of this report ("Some Alternative Reproductive Strategies in Artificial Molecular Machines") a class of artificial molecular machines consisting of folded and interconnected chains of basic artificial molecule constituents is described. Such machines are shown capable of carrying out any effective computational procedure. They can also produce molecule strings which are coded descriptions of themselves. We then describe how such machines might reproduce themselves by 1) templating from a coded description, 2) employing their descriptions as blueprints for the active construction of copies of themselves, or by 3) employing partial or complete copies of themselves as blueprints.

To obtain these results, the strategy is the following. First it is shown how the tape accessing, tape movement, instruction types, and transfer operation of a Wang Universal Turing Machine can be achieved in the molecular machine format. Then a way of implementing, in a molecular machine, the result by C.Y. Lee that Wang Turing machines can describe themselves, is demonstrated. The self-describing tactic is then successively elaborated, the molecular machine capabilities augmented, and the molecule-machine structure re-designed to obtain the results noted.

In Part II of this report (Structural Simplification and Broadcast Communication in Artificial Molecular Machines) we obtain the following principal results: There are artificial molecular machine systems (consisting as before of interacting programs and tape molecules) capable of computation, self-description, construction, and reproduction in which 1) the program molecule can always be cast in *planar* form, 2) the program molecule can always be cast in the form of a *single loop* of constituents, 3) the program molecule can always be cast in the form of a *single string* of constituents, 4) the program molecule (in any graphical form) can communicate with the tape

molecule by means of "broadcast" signals, 5) the program molecule can assume a "dispersed" form (analogous to the freely interacting protein-enzyme chemical machinery portion of a living cell) while the tape molecule remains in the form of a connected string of constituents (analogous to the nucleotide genetic string of living cells).

Part I: Some Alternative Reproductive Strategies in Artificial Molecular Machines

A severely truncated version of this first part of the report appeared in the Proceeding of the Conference on Biologically Motivated Automata Theory (McLean, Va., June 1974).

I.1. Introduction

Artificial organisms are mathematical automata which, although they may not model any particular existing biological organism or system, nevertheless observe a criterion of "biological reasonableness" in their properties. The information transactional activities of such automata must be made precise and this explicitness may serve to bring to the attention and to clarify the conditions which existing empirical systems must somehow satisfy, conditions which may have been left obscure or even gone completel unnoticed in the course of empirical investigations. In addition, artificial organisms can be a fruitful source of *alternative explanations* of observed empirical biological phenomena, and can serve as a vehicle for the *exploration* of broad biological *possibilities*. Artificial organisms may thus be viewed as playing an important role in a theoretical "true biology" more general than usually conceived, a biology which Bernal [1965, p. 97] has suggested "in its full sense would be the study of the nature and activity of all organized objects wherever they were to be found -- on this planet, on others in the solar system, in other solar systems, in other galaxies -- and at all times, future and past."

In this paper we focus on artificial macro-molecular machines. In form these macro-molecular machines are chains (possibly folded and interconnected) of basic molecule-like constituents. The machines are designed so as to possess computational and constructional capabilities, and in this paper we describe how such machines are capable of universal computation, how they can construct molecule chains which are a coded description of themselves, how a machine may replicate by employing its description as a template, how a machine may replicate by employing its description as a blueprint to construct a copy of itself, and how a machine can replicate by directly constructing a copy of itself (without first constructing an externalized blueprint intermediary description of itself). We draw upon the notions of computation and simulation introduced by Turing [1937] and more especially on the application of these notions to problems of machine self-reproduction and self-description as introduced by von Neumann [1951, 1966] and Lee [1963] respectively. (Recent, highly abstract treatments of some of these notions have been presented by Rogers [1967] from a recursive function theory point of view, and by Lofgren [1972] from the viewpoint of recursive function theory in a general systems setting.)

In our treatment we deliberately eschew a detailed mathematical exposition and instead concentrate on making clear the few essential notions employed. In line with this, we will not begin by burdening the reader with lengthy, unmotivated definitions of our molecular machines and their properties but will instead begin with a description of a relatively simple form of molecular machine which we will then subject to successive augmentation and redesign as circumstances oblige.

I.2 Universal Computation in Molecular Machines

We wish first to show that molecular machines can simulate the computations of an arbitrary Turing machine (and can thus carry out any effective procedure). We do this by showing that a molecular machine can carry out any of the actions of a Turing machine as expressed in the Wang [1957] formulation for Turing machines, a system for which the desired universality of computation results have already been shown.

A Turing machine consists of two main parts: an indefinitely extendable tape marked into squares, and a finite state automaton capable of 1) accepting as input the contents of a tape square, 2) modifying the contents of a tape square, 3) moving on the tape one square to the right or left, an 4) changing its internal state. In the Wang formulation, there are two tap symbols permitted, *zero* and *one*, and the precise capabilities of a particul Wang Turing machine finite automaton are expressed as a fixed finitely long program of instructions, composed by drawing from the following list of six basic instruction types: w(0) (write zero), w(1) (write one), R (move right on the tape), L (move left on the tape), H (halt action), t(1,k) (if a one is observed on the tape, transfer to program line named k for the next instruction, otherwise proceed as usual to next immediate instruction on the list).

An example of a Wang program for a Turing machine is:

```
0.  R
1.  t(1,0)
2.  R
3.  w(1)
4.  L
5.  L
6.  t(1,5)
7.  H
```

In this program, the Turing machine will move to the right on the tape until a zero is encountered, at which point it will move one more square to the right and print a one: it will then go left (past the new intervening zero) and continue left until a zero is encountered in the leftward direction. When the zero is found, action then ceases.

We will now show how the actions of any Wang formulation Turing machine can be carried out by a molecular machine, an automaton device all of whose basic constituents and operations are the sort that reasonably may be said to be possible for biological systems at the macro-molecular level.

In our molecular machines, we must somehow represent the tape, the program, and the required interactions of the two, including tape access by the program, movement of tape, program stepping to the next instruction, and transfer implementation.

We can represent the *tape* proper, reasonably enough, by a string of molecules of a sort capable of taking on either of two significant states, which we may call *one* and *zero*, states which might in a physical molecule correspond to the presence or absence of an auxiliary molecule or to one or another of two distinct orientations of a molecule type. The *program* also can be represented as a string of molecules. It is not at all unreasonable for example to imagine that an "active" molecular aggregation could act upon a "passive" molecular aggregation so as to place the second aggregate in a state we chose to call the *one* state, nor is it at all unreasonable to imagine an active molecular aggregation that might place a passive aggregation in a state we call *zero*. Such active molecular aggregations would of course carry out the function of our w(1) and w(0) instruction types. Similarly with the other instruction types, each can be represented by an active molecule or molecular aggregation or sequence, which through a temporary association with a passive "tape" molecule or distinct tape aggregate of molecules, will have the desired effect.

Although, as we have noted, an aggregate or sequence of molecules may be required to produce the desired effect of a single instruction type, we will hereafter speak of these possible macro-molecular instruction complexes simply as molecules, and schematically represent them as single entities. In the operation of our molecular machine we shall require the following.

A string of program molecules will be in sliding contact with a tape molecule string at a single location, and will, if the active program molecule is a w(0), affect the contacted tape molecule to place it in significant state zero; if w(1) to place the tape molecule in state one; if R, will produce a slide to the next tape molecule to the right; if L, will slide to the next tape molecule to the left; if H, will halt action; and if t(1,k) will implement the necessary transfer operation (to be described in more detail later). In addition, if either an R or an L molecule action would result in disengaging the program and tape molecule chains, we will require that a zero molecule be synthesized or otherwise be recruited from the environment and attached to the end of the tape molecule. Finally after each program molecule has carried out its action, the program molecul will relinquish its active status as well as sliding contact with the tape molecule to the next adjacent program molecule, thus "automatically" accomplishing the program-stepping function.

We have said above that the t(1,k) molecule will carry out the necessa transfer operation. This instruction is (unlike the others) *not* "biologically reasonable" in its original form. In the Wang system, if the transfe condition is satisfied (the tape square under scan contains a one), then the next instruction is to be taken from line k, where k is a unique program line name (usually the k-th line from the beginning of the program) In the Wang system (as in programming languages generally) the responsibility for implementing the transfer to the correct location is a property of an assumed, unseen, supervisory system. In our molecular system this assumed mechanism must be given "biologically reasonable" physical implementation.

We do this in the following fashion. For each transfer instruction in a Wang program we substitute a CT (conditional transfer) molecule; before each transfer target point in the original Wang program we insert a new sort of molecule, a TT (transfer target) molecule. We now join the CT and the TT molecules directly together. Such a joining will force the intervening molecule chain into a loop. (Since the number of intervening molecules might be very small, making the loop connection difficult, we can introduce the notion of NOP (no operation) molecules which can be inserted in any desired number to enlarge the loop and to facilitate the transfer connection.) Now if the transfer condition is satisfied, activation is relinquished to the TT molecule and hence automatically to the intended transfer location molecule. If the transfer condition is not satisfied, activation is merely passed on to the original next molecule.

In Figure 1. we show how our earlier Wang program (and an associated tape molecule) might appear as part of a larger molecular machine. Note that although it is possible for this schematic example program molecular machine to be placed in planar form, very complex such machines may assume non-planar form.

By these techniques, the actions of any Wang system can be realized in a molecular machine; we can thus claim our first result; namely that for every effective computational process, there is a molecular machine which can carry out the process.

We will implicitly employ this result frequently in much of what follows. It should be pointed out that the potential universal computational power of our molecular machines is not strictly necessary to obtain most of the results which follow. The property does however relieve us of the chore of providing numerous separate proofs of the computational powers we shall from time to time require our molecular machines to possess.

Having completed the design of the first of the kinds of our molecular machines, let us stop and ask ourselves how such a machine may reproduce itself. One possibility is obvious: that a simple self-templating, a molecule-for-molecule matching, might serve to produce a second molecular

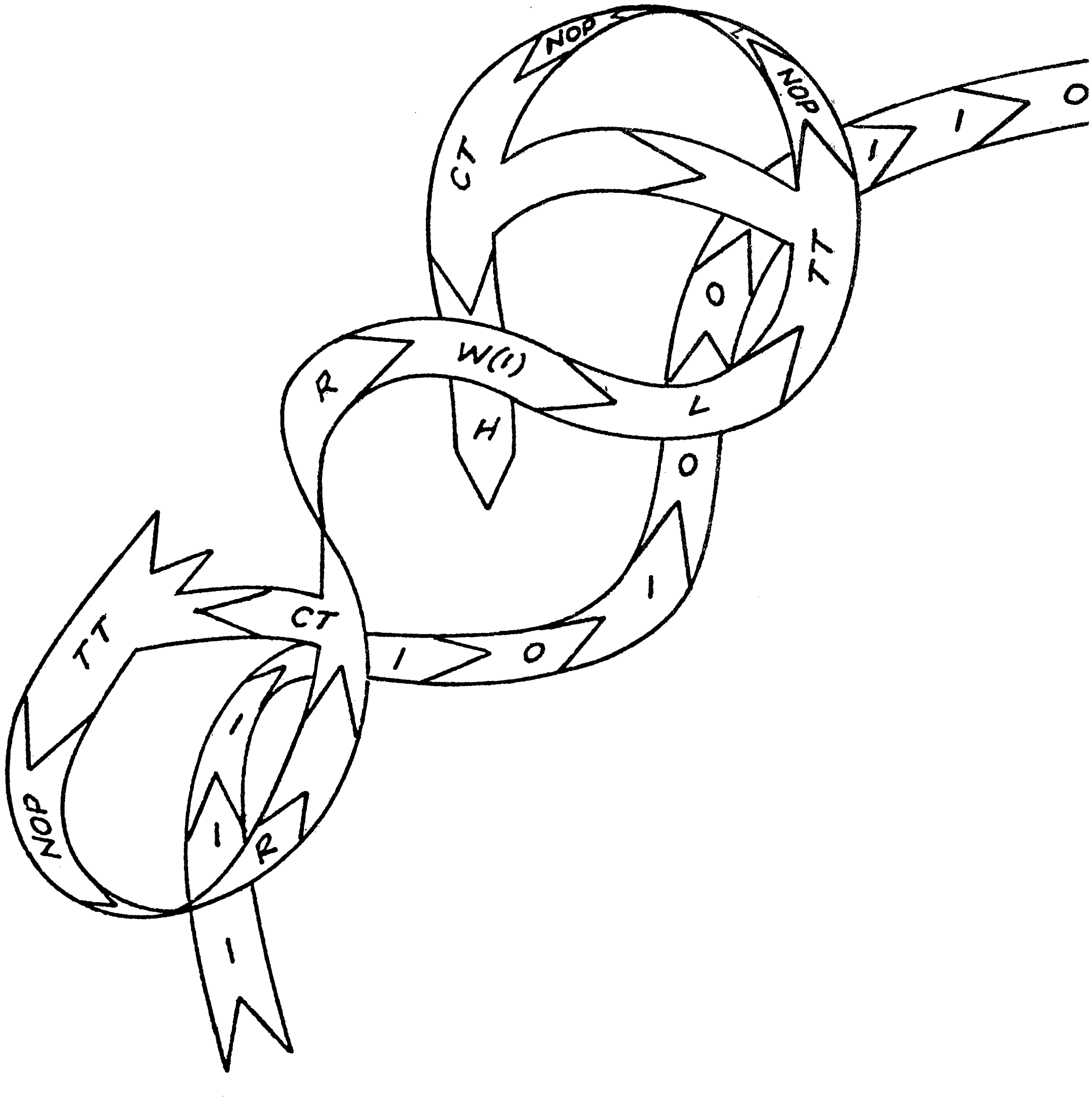


Figure 1

Figure 1. Active Interconnected Program Molecule and Associated Passive Tape Molecule

machine identical to the original. This suggestion suffers from several disabilities: 1) many machines of interest and importance (including universal machines) may be quite complex structurally, possibly assuming non-planar form, and thus any simple templating procedure would (in the non-planar cases) require breaking and re-connecting of what would be co-valent molecular bonds, in order to effect the complete disengaging of parent and offspring machine; we have as yet introduced no mechanism by whi such breaking and re-connecting can be carried out. One way of resolving this would be to unfold the parent machine, copy it, and then place parent and offspring in the folded state once more; as yet however we have provide no means for accomplishing this, either. 2) We have as yet not provided a *source* of the molecular constituents, no mechanism for producing molecules of the appropriate *kind* and *amount*, and no mechanism for their transport an positioning.

In short, in our system in its present form reproduction can be secure by direct templating only at the cost of several substantial unjustified assumptions about the capabilities of our machines or of the environments in which the machines reside.

I.3 Self-Description in Molecular Machines

Lee [1963] (inspired in part by ideas of von Neumann [1951,1966] first set forth in 1948) proved the existence of self-describing Turing machines, and Thatcher [1963] worked out the complete design of one such self-describing machine.

Since our molecular machines can carry out any Turing computational process, and the Lee-Thatcher technique of self-description is only a special sort of computation, it follows that our molecular machines can also be self-describing. Since we wish to build upon this notion of self-descri tion to demonstrate several methods of reproduction, and because in both describing and reproducing we wish to limit our mechanisms to actions which can be phrased in a "biological vocabulary", we will pause to explain the essential features of the Lee-Thatcher self-describing results.

Both Lee and Thatcher employ the Wang program Turing machine concept in showing their self-description results, and we will do likewise. Recall that the six Wang program instructions types are: w(0), w(1), R, L, H, t(1,k). For explicitness we fix upon the following coding: w(0) = 01, w(1) = 011, R = 0111, L = 01111, H = 011111, t(1,k) = 0111111(1^k). Thus, each instruction type is coded by unique number of ones, preceded by a zero; the transfer instruction is coded by six ones, the number of *additional* ones giving the transfer target lines. A complete *coded program* thus would consist of a string of such coded individual instructions, viz. a string composed of sequences of ones separated by single zeros. A *self-describing program* is one which, acting upon an initially blank tape, will print out a string of zeros and ones which according to the agreed upon coding, represents the program itself.

In the method of self-description we shall present, the program to be described consists of two principal parts, which we shall call the *controller* and the *genetic record*. The controller will be a finitely long program segment which has the capability to carry out some straightforward decoding and interpreting functions. (The controller part of the program may optionally also contain an additional finitely long program of instructions for carrying out supplementary computational activity not essential to the self-describing process; we shall have occasion later to employ this property.) The genetic record part of our self-describing program will take an especially simple form. It will consist of a list of instructions of the w(0), w(1), and R type only. Such a program can clearly, at most, print out a sequence of zeroes and ones.

Now our controller program segment itself can be represented, accordin to our coding convention, by a sequence of zeroes and ones. We construct the genetic record program segment so that the sequence of zeroes and ones it prints out will be exactly the agreed upon coded description of the enti controller part of the program. (This genetic record part of the program will of course be several times longer than the controller part, for each instruction of the controller will require several instructions from the record to produce the coded description.)

The complete process of self-description can now be described. We begin with the genetic record part of the program. This part of the program proceeds to print out a description, according to the agreed upon coding convention, of the controller part of the program. Thus, on the tap we will then have a description of part of our program, the controller, but as yet no description of the genetic record part of our program. At this point the controller part of the program takes over and begins to examine the sequence now on the tape. A description of the genetic record can be constructed by an analysis of this sequence. For example, a *zero* must have been a consequence of a w(0) instruction; a w(0) is described by 01, which can be printed by the controller on the end of the sequence. If a *one* follows, then it must first have been made possible by an P instruction, so that 011 must first be appended to the existing printed sequence, followed by 01, for the w(1) which printed the one, etc.. By this means (easily within the power of a finite program, and in particular of our arbitrarily complex controller), a description of the genetic record program itself is recoverable from the sequence of zeroes and ones which the record caused to be printed on the tape in the first place. We thus end with a complete description of both controller and genetic record part of the program, and self-description has taken place. (In any attempt to produce an actual self-describing program there are a few niceties which must be observed about line numbering conventions, transfers and their locations, etc. which if spelled out would serve only to obscure the simple essential features of the process. See Thatcher [1970].)

Now clearly the same self-describing technique can be employed in our molecular machines. (We shall of course have to augment the coding convention to include descriptions of our additional TT and NOP molecular elements.) Notice however, that there is one distinct deficiency in this self-describing technique, especially if we wish to preserve "biological reasonableness". The designation of transfer target location by means of a naming and counting convention is very "unnatural". This unbiological feature which we exorcised from the molecular machine itself has returned in its coded description.

We can however show that there exist self-describing molecular machines whose descriptions remain biologically reasonable. We do this by introducing the notion of unit size rigid molecular constituents with rigid bonding angles, and fixed bonding distances. We also introduce some additional kinds of null-behavior rigid constituents, these new null-behavior constituents will implement right angle turns in three-space. With structural rigidity, and these new "corner turning" elements, the molecular machine structure and consequently the transfer points, will be determined by the constraints of the molecular constituents themselves. Note that by an appropriate use of shape forming nulls, the molecule chain can be forced into spatial routing which will safely avoid undesired clashes with other parts of the chain, and which can facilitate contact with the *tape* molecule chain (which may also be composed of rigid constituents). Figure 2 shows how our original example program might appear in rigid constituent form.

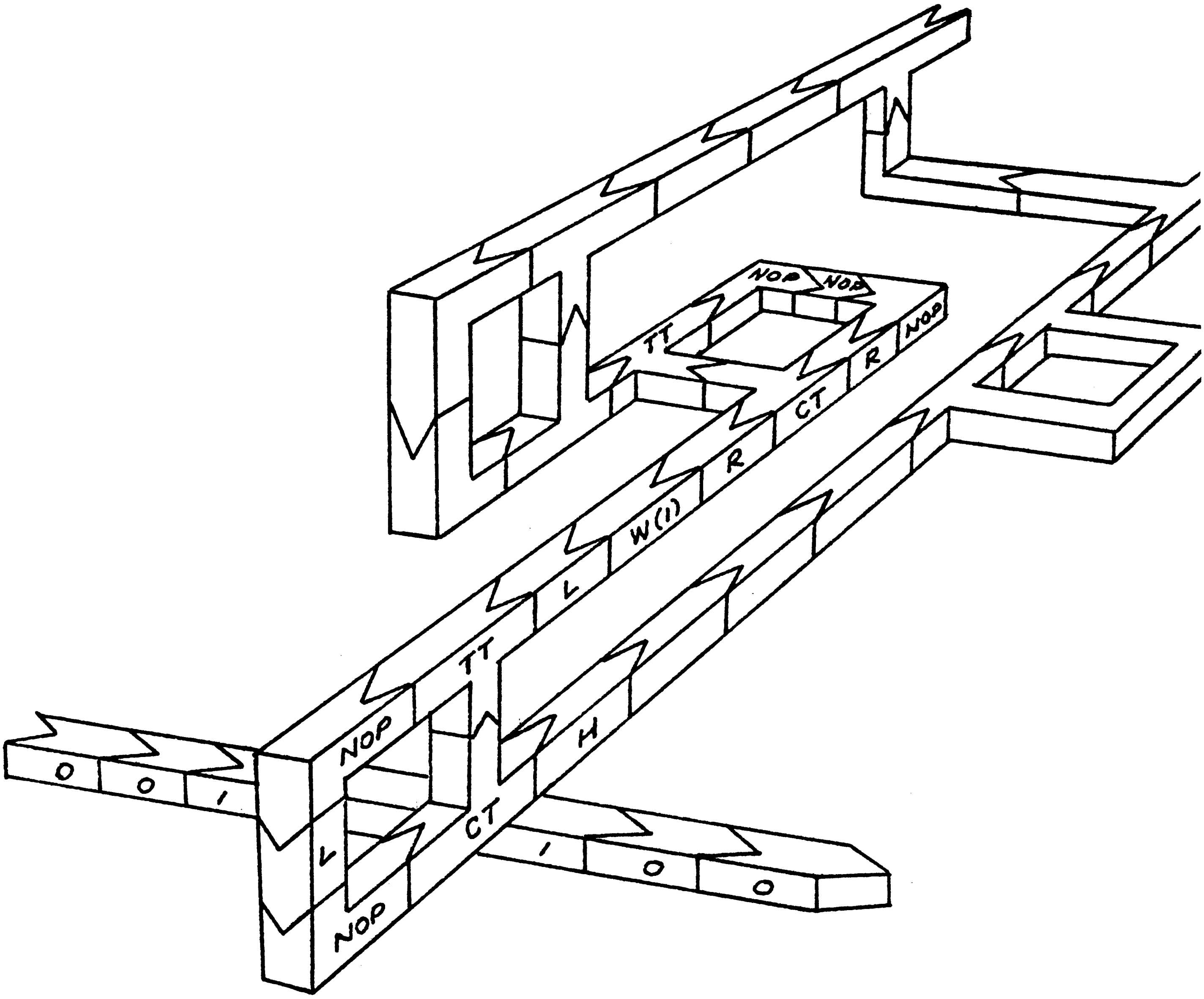


Figure 2

Figure 2. Active Program Machine and Passive Tape Machine Composed of Rigid Constituents

A description of such a "rigid" molecular machine can then be the serial (primary structure) listing of the constituent molecular types. With this, the molecular machine itself will be recoverable from its description, and both machine and description will observe "biological reasonableness". Also note that in describing our molecular machines we may employ a *fixed-length* coding convention, the length of each of our code blocks dependent on the total number of elementary constituents we shall ultimately require to be coded by a zero-one sequence.

I.4 Reproduction in molecular machines templating from descriptions

Since the description of our molecular machine contains all the information necessary for a complete specification of the molecular machine the question arises whether the description information can be employed in a "biologically reasonable" fashion to creat a copy of the original molecular machine and so to display self-reproduction.

If we assume that the environment can supply a population of the basic molecular machine constituents, (possibly temporarily accompanied by additional constituents to facilitate transcription), reproduction might be attained by pairing between the molecules of the coded description and the molecular machine constituent types. As the proper molecular constituent sequence is formed, the molecular machine string of constituents peels away, the structure forming properties of the primary sequence of molecular constituents assuring the proper three-dimensional structure. By this means an exact copy of the original molecular machine can be produced.

If, in order to complete the reproductive process, we should like to assure separation of original and offspring (so that the two may be subjected to different environments, and undergo different histories) we might suppose the original machine is enclosed in a membranous envelope, through which the original machine, because of its folded structure cannot pass. The *description* macro-molecule however, because of its flexible string-like structure *can* pass (along with individual molecule constituents)

The complete template reproduction technique would then be as follows: The controller segment of the original molecular machine begins by synthesizing a population of molecular constituents. These detached and dispersed constituents insinuate themselves through the one-way membrane. The molecular machine then produces a description of itself. This description, also, being in string-like form, can pass through the enclosing membrane. The free molecular constituents form themselves against the description and, connected, peel away, taking on the structure required by virtue of the self-assembly properties of the constituents themselves, and the reproduction is complete.

At this point there remain many unresolved problems, some of which we will address in the next sections.

I.5 Synthesis of Constituents and Construction of Molecular Machines

Among the assumptions we have made (but not accounted for) is that the molecular machines themselves might synthesize individual molecular constituents of the very sort they themselves are constructed of, and even possibly construct simple molecular structures which might be useful or necessary to facilitate the templating process. We now discuss how this synthesizing and constructing process can take place in our molecular machine format.

Recall that our controller segment may include a finite amount of molecular machinery not strictly committed to the direct self-descriptive process. There is no difficulty in incorporating such additional molecular machinery, along with the essential self-describing machinery, in the molecular machine. This additional molecular machinery could have the task of synthesizing molecular constituents. One way in which this might take place is the following.

Our system already provides for the recruitment of "zero" molecules from the environment whenever an activated R or L molecule reaches the end point of a molecular "tape" string. We now introduce the notion of a new basic molecule type, the S (for Synthesize) molecule. This molecule being brought into active association with a passive molecule will cause the passive molecule to change its state (in the same fashion as a w(1) molecule causes a zero-molecule to become a one-molecule). Successive associations with activated S molecules, will produce successive changes in molecules with which the activated S-molecules are in temporary association Thus, a sufficiently long sequence of associations with activated S molecules will cause initial zero molecules to become any molecule (including S-molecules themselves) of our repertoire of basic constituents.

This means that we can have a constituent-synthesizing region of our molecular machine, in which basic constituents are manufactured and release into the environment. Also note that in order to implement constituent release, we shall also have to introduce a "detachment" molecule to our repertoire, a molecule which activated and in temporary association with a molecule, will cause the severing of the structural attachment that molecule has with its predecessors.

It is clear that a program can be devised which will produce and release large numbers of the basic constituents necessary for the reproductio by templating from descriptions of molecular machines. One technique in particular suggests itself. The molecular machine begins by examining the description of itself. It can note one-by-one what sorts of molecules are required, and, as each type is ascertained, it can transfer to a subroutine for producing a molecule of that type. By this method, exactly the constituents required could be produced and made available for the templating process. If some degree of redundancy is desireable, the molecular machine might make two, or ten, or more of each kind.

Since in this technique the description is examined in sequence, and the basic constituents can be produced in the order of their appearance, the possibility that the reproduction of a molecular machine might be accomplished directly from the description without recourse to templating at all arises.

These techniques are discussed in the next section.

I.6 Reproduction: Construction Under Control of the Molecular Machine

In the last two sections we described how a molecular machine could reproduce itself by producing both a description of itself as well as the requisite number and type of constituents, releasing the description and constituents, so that they might form, by templating at a removed location, a copy of the original machine. These processes can be employed to carry out a reproduction more directly.

The technique is as follows. First (as discussed earlier) the molecular machine produces a string of molecules which is a description of itself. Referring then to this description, the molecular machine, determines one-by-one the type of constituent required, then retreats to the end of the description string and synthesizes the required constituent. By continued repetition of this process the molecular machine will finally end with a sequence of molecules, the first half of which is its description in zero and one molecule types, and the latter half of which is an exact copy of the molecular machine constituent primary sequence.

The detachment molecule can now be employed to sever the connection between the description molecule and the new molecule machine constituent sequence. (We shall also want to be able to communicate an "active" status to a region of offspring molecular machine, else offspring would possess no autonomous self-reproducing properties.) During construction, and upon release, the presence in the primary constituent sequence of structure forming properties will by self-assembly impose the requisite three dimensional form. (It should be noted that, above and later, the readings back and forth will require some sort of "punctuation" marking the end of the most recent molecule segment read, the end of the entire segment ultimately to be read, etc.. Such punctuation can be implemented either by employing the notion that not all of the molecules present shall denote functional components of the machine proper, some portion of them -- perhaps every other one -- being available to serve as markers, or we can introduce some new primitive properties into our system, permitting the placing and removing of markers.)

Since our molecular machines can now construct other molecular machine we now possess a way of producing not just individual molecules, but can produce whole molecule aggregates. Therefore, if we wish to persist in an explication of template modes of reproduction, a much more sophisticated form of this approach can now be implemented. Instead of (as in Section 5) merely synthesizing and setting free large numbers of the required kinds of molecules, we can produce for each molecule type, a whole molecular adaptor complex designed to facilitate the templating-transcribing process. We might for example wish to construct a chain of molecules which has at one end a sequence of zero-one molecules which is identical (or if desired, complementary to) the zero-one descriptive sequence for particular molecule types appearing in the description chain of molecules for the entire machine. At the other end of the constructed chain we might have an attached molecule of the type described, oriented in such a way as to facilitate its bonding with the molecules of facilitating chains afixed at adjacent description locations in the complete description sequence.

Such "transfer-RNA-like" structures can be produced by having the original molecular machine read the description it has produced to first determine the kind of molecule to be synthesized. The machine then retreats to the end of the description molecule sequence and constructs a "mini-molecular machine" which itself will be able to perform some simple construction tasks -- essentially to synthesize or recruit a molecule of the desired described type. (We do not have the *original* machine form the desired molecule directly because then the new molecule would be an integral part of the facilitating aggregate, not merely temporarily attached to it.) At the opposite end of this auxiliary structure, the original machine will create a description matching sequence of zero and one molecules. The original machine will then activate the offspring minimachine and detach it from the end of the description molecule sequence.

The original machine can of course produce one or more of these facilitating aggregates for each of the kinds of molecules described in the original zero-one sequence.

I.7 Reproduction without use of a Description Intermediary

In all the forms of reproduction so far described, we have made use of the ability of molecular machines to construct and then read complete *descriptions* of themselves. We now show how molecular machines can reproduce themselves in a more directly constructional fashion.

It is clear that instead of producing a *description* of the controller unit, the genetic record of our molecular machine might be made somewhat more sophisticated, and produce, directly, a *copy* of the controller unit. In other words, the genetic record, instead of containing a series of w(0), w(1), and R instructions, yielding a sequences of zeroes and ones describin the controller sequence, will contain a series of w(0), and S, and R instru tions which will cause the synthesis of each constituent in sequence. Whe the genetic record ceases its activity, it will have synthesized a sequence of constituents which is a *copy* of the primary constituent sequence of the controller unit of the original molecular machine, not merely a descriptio in the zero-one molecule "language".

We must now complete the reproduction by constructing the genetic record constituent sequence. We have heretofore accomplished this by havin the molecular machine "read" and interpret the *description* of the controlle in order to ascertain the genetic record program which must have produced it. We have as yet however provided no means by which a molecular machine can "read" molecules other than those of the descriptive zero and one sort (our conditional transfer molecule is the only one which can ascertain molecule types).

If we design our molecular machines appropriately, they can possess th analytic capability to examine molecular constituents (constituents of the same sort of which they, themselves, are composed) and ascertain their type Indeed, the analysis could without harm be "destructive" since once the constituent is identified, a synthesis procedure could restore the subject constituent to its original status. (That this sort of analysis and restor tion is possible by a sufficiently complex molecular machine, is evident. It is clearly analogous to the actions of a laboratory technician following a set of rote analysis procedures for the contents of a rack of test tubes, where the possible range of contents is known, and where once the contents are determined, the tube is to be correctly labelled and refilled. That the tube may contain constituents of the same sort as may be employed in the analysis procedure, and indeed in some cases of the same sort of which the technician himself is composed, does not effect the case.)

Any particular analysis procedure to be carried out by our molecular machine on a copy of itself (or a copy of a part of itself) will of course usually be a function of the ways in which any particular molecule types may be derived from others, and converted in turn to others. Since in an *artificial* system of molecular machines we can *assign* these relationships (as long as they are biologically plausible) we might, for definiteness of example, fix upon a constituent synthesis process whereby successive S molecule active associations (beginning with a *zero* molecule, say), produce each of the other molecule types in a fixed known sequence *eventually returning to the initial constituent type*. Thus, in order to ascertain the type of an unknown molecule, we can subject the molecule to a "decoding" sequence of S molecules. After each of the S molecules, we place a conditional transfer molecule (which can detect a *one* molecule if such is ever produced in the course of the analysis).

The number of S molecule associations required to convert a given molecule into a detectable *one* molecule, will reveal its type. When the transfer condition is satisfied, we are routed to a region of the molecular machine controller where, first, by a series of S molecule associations we restore the molecule under inspection to its original form, then enter into a part of the controller program which will construct that segment of the genetic record which must have been required to produce the newly ascertained molecule type. Finally we return to examine and test the next molecule in line. All the above can be effectively carried out.

At the conclusion of this process we will end with a complete constituent chain of molecules, which, in its active three-dimensional form (which it must assume by virtue of its own constituents) will be an exact copy of the original molecular machine.

I.8 Simplified Reproduction

Since we now possess a means by which a molecular machine can "read" and act upon another machine, a new, simplified method of self-reproduction becomes possible. In this method we begin with a *pair* of associated molecular machines, each of which possesses the sort of properties we have ascribed to the controller segment of our earlier molecular machine system. One of these two controller units is initially "active" and can act upon its associated structurally identical mate. This first controller will read its mate (molecule by molecule) twice and synthesize two copies of the mate. We end thus with four copies in place of the original two, and so in rudimentary sense complete self-reproduction has taken place. This rudimentary self-reproduction ignores some of the specific problems of activation, detachment, and alignment however.

A somewhat more detailed description of this reproduction process is thus as follows. The first controller unit reads its mate and synthesizes a copy of the mate, attaching the new constituents at one end of the existi mate. Once a copy of the mate is produced, activity is relinquished by the first controller to the original second controller or to its new copy. This newly activated machine can now read the originally active first machine, and furthermore produce a copy of it. At this point, we thus have four identical controller units (in place of the original two) in the form of two associated "double" machines. Self-reproduction can now be complete by shifting the activated region *and* the physical association to the point where the four machines join. The activation can then be divided, assigning it to one of the original pair and to one of the newly synthesized pair of machines. Each active machine can then cleave the connection between its opposite pair of machines. We end thus with two separate pairs of associated identical machines, one element of each pair being active, one passive, and reproduction is complete.

Finally, there is a very important and very compact variant of this last described logical process of reproduction, a variant which corresponds closely to what is thought to be the actual state of affairs in existing biological organisms. In this variant we begin with an active program controller machine and an associated passive description machine (which can be in our zero-one molecule language, and in simple string form). The passive descriptive string machine spells out the primary structure serial order of constituents of the active machine with which itsis associated.

In action, the active machine (playing the role of the cytoplasmic enzyme machinery) reads the passive machine (which plays the role of DNA) and produces first another copy of the passive machine, then another copy of the active machine (i.e. a copy of itself). Since we begin with a pair consisting of an active and passive machine and end with *two* such machine pairs, reproduction has taken place.

I.9 Summary of Reproductive Strategies

We now summarize the principal techniques we have described, by which molecular machines of the sort discussed can reproduce themselves.

1. Given: a molecular machine consisting of a controller and a describer of the controller. The describer produces a description

of the controller. The controller, from this information, deduces a description of the describer, and prints out this description. This complete description of the machine can then, by use of complementary molecules, templating with and without the use of facilitating aggregates, etc. be employed to produce a copy of the original machine.

2. Given: A molecular machine consisting of a controller and a descr of the controller. Employing the method of 1., a complete description of the original machine is produced. Reading this complete description (or following its implicit instructions), the controll *constructs* a copy of the complete original machine.

3. Given: a molecular machine consisting of a controller and a *constructor* of the controller. The constructor produces a copy of controller. Either the original or newly constructed controlle now examines the other controller and from this examination deduce the structure of the constructor and produces the constructor, completing the self-reproduction.

4. Given: a molecular machine consisting of a *pair* of *controller* units, one initially active, one passive. The *active* controller unit of the pair inspects the passive unit and produces a copy of it. The originally active unit now relinquishes activity to the newly constructed unit, which proceeds to copy the originally active unit. (Alternatively, the originally active unit could itself employ a passive copy to produce a second active unit.) This yields *two* connected pairs of controller units, which when separately activated and disengaged, completes the reproduction.

(Alternatively, given an active controller machine and its passive *description*. The active machine inspects the passive machine and produces a new passive machine and a new active machine, completing reproduction.)

I.10 A Critique and Some Further Directions

In the molecular machines presented here the program molecule and the tape molecule are required to be in sliding contact and there may be the impression that the required interaction would be difficult or even in some cases impossible. This appears especially so in the exaggerated "rigid plumbing" of Figure 2 where it is easy to imagine that the program and tape molecules might intertwine and interlock in ways that would block or impede the sliding contact. Since in molecular machines we undertake to confine the properties employed to those which are physically plausible the problem deserves to be faced: we have accepted an obligation not to stray too far from the recognizably empirical realm.

In mitigation it should first be pointed out that the "plumbing" diagram of rigid elements is more stylized than it need be, making the problem seem worse than it is. We can, for example, add many additional NOP (no operation) elements so as to bend and fold the structures to facili contact. In particular we can force the molecule program structures to be, roughly speaking, "convex", so that interior structural traps and cul-de-sa can be avoided and critical communication (and especially transfer) regions are maximally exposed on the outer surface. Next, we note that *permanent* rigidity of bonds and constituents is not necessary. (In the cas of the tape description molecule, rigidity is not required at all.) For th molecule, *once the secondary bonds have formed*, the rigidity of all the bonds can be relaxed, as along as all the interconnections are retained. This "semi-denaturing" will permit a structural flexibility which will facilitate program molecule and tape molecule interaction.

Yet another solution to the problem lies in the simplification of program molecule structure. At the cost of increasing somewhat the complex ity of individual components (though they will still remain biologically plausible) any program molecule can be recast in simple circular or even straight string segment form. (Wagner [1967] presents several relevant structural simplification procedures.)

Another possibility that suggests itself is also, biologically, the most familiar and likely: the active and passive parts of the molecular machine system can communicate by enzyme catalyzed reactions through a fluid medium contained within an envelope. This can be carried out in artificial molecular machine terms providing we permit a sufficient number of distinct enzymes and distinct sites of activation. (See Laing [1973] for a brief discussion of the implementation of such a "free-floating" automaton system in a biological context, and Holland [1974] for a carefully worked out formalism for such "broadcast" communication automaton systems.) This enzyme communication, while biologically more realistic, is also for the logical analysis of self-reproduction much more challenging. The numerous enzyme communicatory substances and active sites, as well as the fluid transporting medium and its containing envelope which are required to replace the relatively straightforward notion of direct contact, must now play such integral roles in the self-reproductive process that their creation logically should be included in the self-reproduction loop. That is, the enzymes, the fluid, and envelope are part of the active molecular machine, and they must have their coded counterparts in the passive description molecular machine; also the active machine must, by reference to the passive machine, create new active and passive copies of them for use in the next generation.

In Part II of this report we pursue some of these matters further.

Part II: Structural Simplification and Broadcast Communication in Some Artificial Molecular Machines

II.1 Introduction

In the first part of this report we showed that for any Turing machine cast in the Wang program machine system, there is an artificial molecular machine system consisting of an interacting pair of artificial macro-molecules which can carry out the same computation. In addition we described how such artificial molecular machine systems might produce descriptions of themselves, construct other artificial molecular machine systems, and reproduce themselves.

Of the two interacting artificial macro-molecules of which these systems consist (the program molecule and the tape molecule) some program molecules may be topologically very complicated; and, particularly in some of the reproductive schemes discussed *both* program and tape molecules may be topologically very complex.

This notion of two topologically complicated molecules in sliding contact along their full length places a heavy burden on physical realizability. What we propose to do in this second part of the report is to mount an independent *two-pronged* attack on this putative deficiency of the presently conceived molecular machine system.

We will first describe some techniques by which the structure of the program molecule of our artificial molecular machine system can be immensely simplified; such structural simplification would greatly facilitate the sliding contact between the program and tape molecule.

In the second prong of our attack, we show how we can (by employing the notion of "broadcast" communication) eliminate the requirement for direct sliding contact entirely, thus obviating completely the original difficulty.

More particularly the principal results of this second part of the report are that there are artificial molecular machine systems capable of computation, self-description, construction, and reproduction in which

1) the program molecule can be cast in *planar* form, 2) the program molecule can be cast in the form of a *single loop* of constituents, 3) the program molecule can be cast in the form of a *single string* of constituents, 4) the program molecule is free to assume any of the structural forms and in which communication between the program molecule and the tape molecule takes place by means of "broadcast" signals, 5) the program molecule can be in "decomposed" form (analogous to the freely moving and interacting protein-based active machinery of the living cell) while the tape molecule remains in the form of a simple connected string of constituents (analogous to the cell's nucleotide genetic componentry).

In order to obtain these results, fundamentally we need only show that our re-structured artificial molecule machines systems possess computationa powers equal to our earlier systems. The self-description, construction, and reproductive properties will follow by the reasoning employed in Part I In order to obtain our new results, we will be forced to pay a (modest) price in increased complexity of basic individual molecule constitutents, along with some assumptions about transport and decay of molecular material

II.2 Planarity of Program Molecule Structure

In this section we show that the topological complexity of the progran molecule can be greatly reduced; indeed we show that graphical form of the program molecule can always be made *planar*. (This result has the followin significance: not only is sliding contact between program and tape molecul facilitated by the simplified program molecule structure, but the program molecule can be replicated in template fashion, without entailing the break and re-constituting of primary connections between molecule constituents -- and thus not requiring as strong an assumption of automatic self-assembly c constituents.)

The troublesome graphical complexity of the program molecule is imposed by the (unavoidable) requirement for a means of *transfer* of activation and contact in any very general computational activity. In our original conception (Part I) if a conditional transfer constituent is activated (and the transfer condition is also satisfied) then activation and contact is routed along a (possibly auxiliary) molecular constituent chain (rather than to the next constituent of the primary chain) to a transfer target location at a removed point in the primary constituent chain. Since no constraint is placed on the location of conditional transfer constituents and their transfer targets, the program may assume arbitrary (including non-planar) graphical form. (See Figure 3)

To show that any program molecule can be re-cast in planar form, we begin by first routing each of the conditional transfers *not* directly to their distinct ultimate transfer target constituents, but instead to a single junction point at the terminal end of the primary constituent sequence of the program molecule. (This can of course be carried out with no crossing of constituent chains.)

We then append to the terminal end of the program molecule a special transfer implementation segment of constituents. This segment consists of a sequence of conditional transfer constituents, from which, by means of auxiliary chains of constituents, transfer of activation and contact can be made back up into the primary chain. Now if the transfers out of the appended transfer implementation segment can be so arranged that the transfer from the *lowest* point in the newly appended transfer segment goes to the *highest* point in the primary chain, and the next lowest to the second highest, etc., all of the transfer chains of constituents out of the segment can be *nested* so that they need not cross, and thus will not violate the planarity condition.

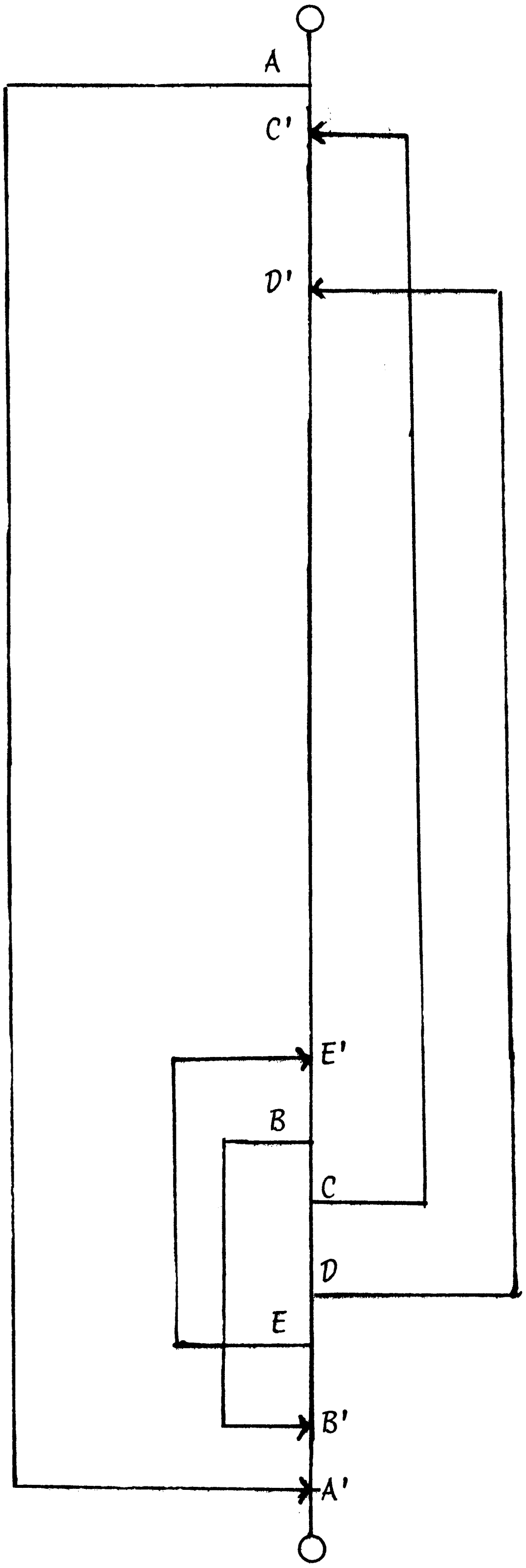


Figure 3

Figure 3. Schematic Structure of a Program Molecule. The conditional transfer requirements (initiated at points A, B, C, D, E and terminating at A', B', C', D', and E' respectively) may impose a non-planar structural form on the program molecule.

What remains is to show how we can match each original transfer origin point to its correct exit point from the transfer segment. We accomplish this as follows. In each of the auxiliary strings from an original conditional transfer constituent to the head of the appended transfer segment we insert a sequence of constituents which superimposes a of special marks on successive constituents (moving to the right) of the associated *tape* molecule. The number of marks imposed on this external medium will signify the position of the transfer from within the appended transfer segment, back to the intended ultimate transfer target in the original primary sequence of program molecule constituents.

To implement properly this ultimate transfer from the appended transfer segment, the transfer segment will consist of a string of constituents which test first to see if the tape molecule constituent with which it is in contact possesses one of the special superimposed marks. If the molecule is in a special marked state, the special mark is removed, contact is shifted to the next tape molecule on the left, and, (in the transfer segment of the program) activation and contact is shifted to the next lower constituent If a constituent of the transfer segment finds that the tape constituent with which it is in contact is *not* in a special marked state, then a transfer is implemented (by routing activation and contact along the auxiliary chain of molecule constituents with which each transfer constituent is associated) to the ultimate transfer target in the primary chain. (Figure 4 illustrates schematically the result of converting the program molecule of Figure 3 to planar form.)

Thus in brief summary, the process is as follows: upon the original conditional transfer condition being satisfied, a number of marks signifying the proper position in the appended transfer segment is imposed on an external readable medium (in this case the associated tape molecule); activa

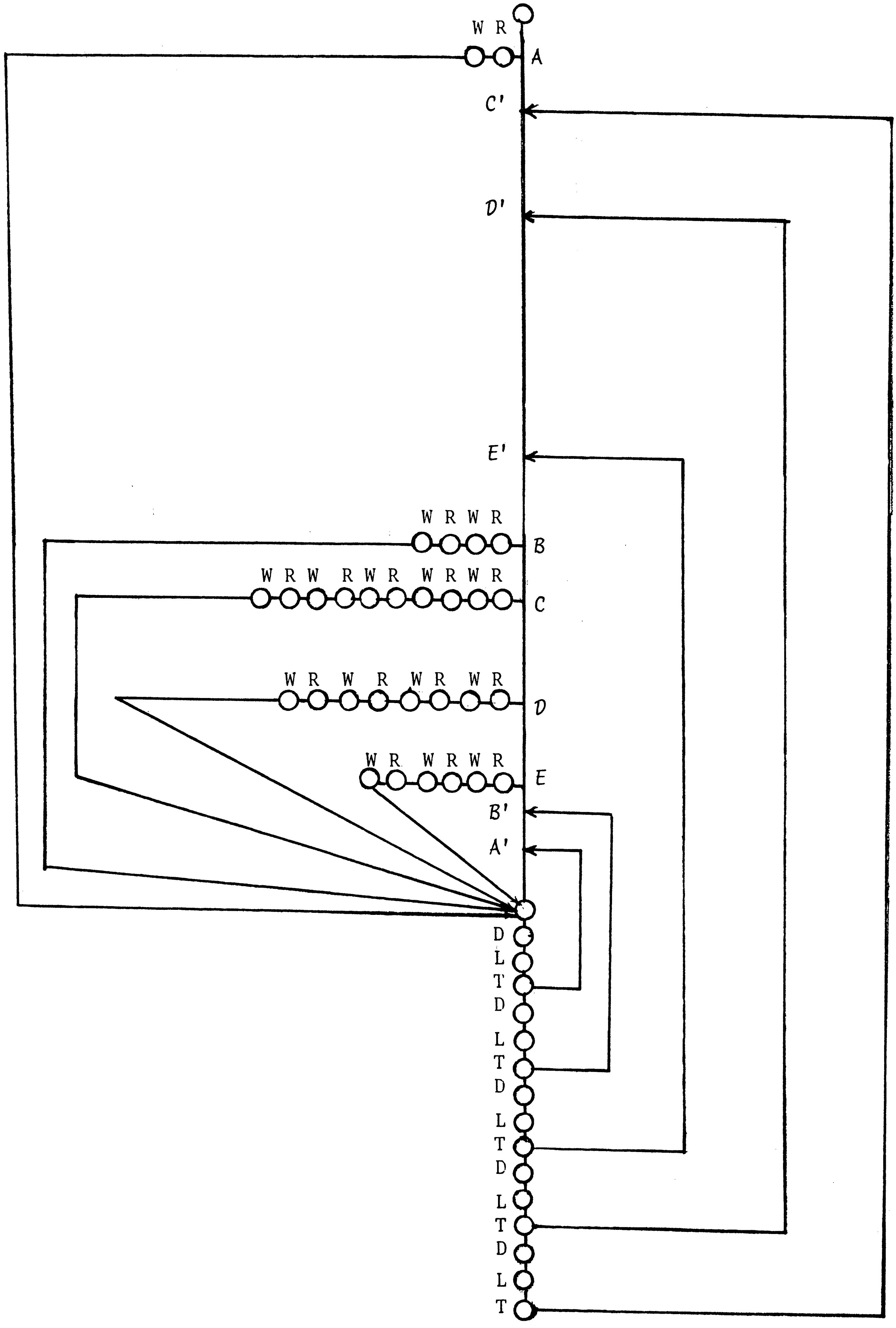

W R
A
C'
D'
E'
W R W R
B
W R W R W R W R W R
C
W R W R W R W R
D
W R W R W R
E
B'
A'
D
L
T
D
L
T
D
L
T
D
L
T
D
L
T

Figure 4

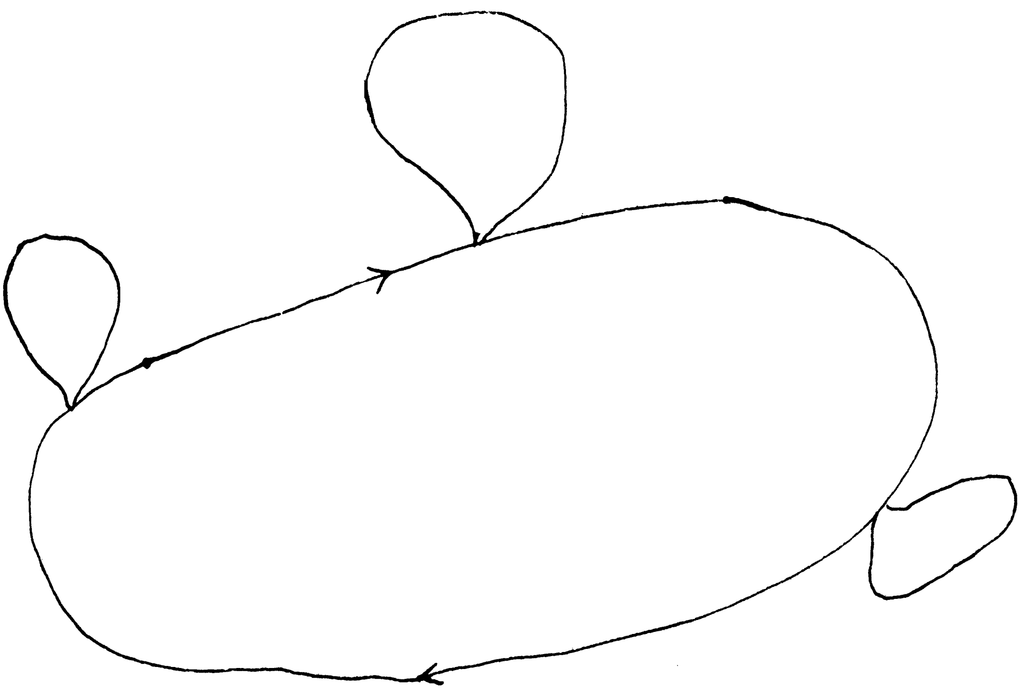
Figure 4. Planar Form of Program Molecule. The program of Figure 3 has been converted to planar form by employing W constituents and R constituents to superimpose special marks on the tape, then routing contact and activation to the head of the newly appended transfer region. In the transfer region the special marks are removed one by one by D (delete) constituents, moving to the lef until none remain. When all marks are removed from the tape, the conditional transfer constituent becomes operative and conta and activation is routed to the proper target location in the main program molecule.

and contact now shifted to the head of the appended transfer segment, the transfer segment apparatus tests for and removes these special tape marks one by one until all are gone; at this point in the transfer segment a conditional transfer becomes operative and activation and contact are carried to the proper transfer target location in the program molecule. Note that at the satisfactory completion of this transfer process, all special marks will have been removed from the tape molecule, and the original point of contact with the tape molecule restored. (This whole strategy of program molecule modification is an adaptation of a tactic employed by Wagner (1967).)

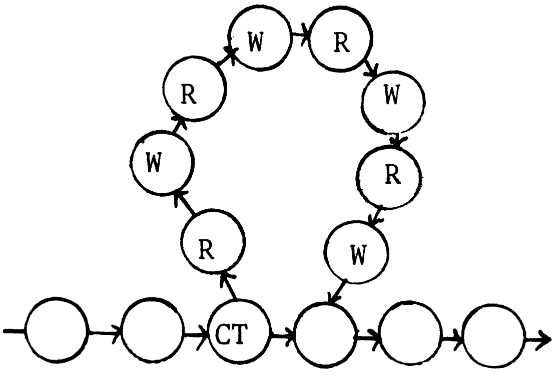
By the described procedure, *any* of our original program molecules can be re-cast in a planar structural form, and thus systems consisting of such planar program molecules and associated tape molecules can (by our earlier argument) carry out any effective computation. Also by earlier arguments there exist systems with planar program molecules which can produce complete descriptions of their own program molecules, and suitably augmented with the ability to sense and synthesize basic constituents, can reproduce themselves in any of the manners described in Part I.

II.3 Further Simplifications of Program Molecule Structure

In this section of the report we will first show that all the computational and constructional results (including self-reproduction) of our earlier artificial molecular machine systems can be obtained in a system in which any program molecule structure can be cast in the form of a *single loop* of constituent molecules to which may be *appended smaller loops* (to be employed in implementing the conditional transfer process). Figure 5 is a schematic illustration of such a program molecule.



a)



b)

Figure 5

Figure 5. a) Schematic Diagram of Program Molecule Structure, b) Detail of Appended Transfer Loop Region.

In this revised program structure, each conditional transfer constituent has associated with it an appended finite loop of constituents. The loop will consist of program constituents which mark tape molecule con: and which cause moves to the right on the tape molecule. The transfer strategy is as follows. Each time a conditional transfer condition is satisfied, activation is routed through the appended loop of constituents (if the condition is *not* satisfied activation proceeds to the normal (next) constituent in the main program). By means of the appended loop, a series of special marks is superimposed on successive constituents of the tape molecule . (The original tape constituent states are *not* "blotted out".) When activation and contact is returned to the main program, a main program constituent will if it confronts a special mark on the tape, *not* carry out its usual computational actions, but will instead *remove* the special mark, move left on the tape molecule, and route activation and contact to the next main program constituent. The consequence of this will be that ordinary main program computational activity will be postponec until a point is reached where all special marks have been removed. At thi point, the main program will be confronting the same tape constituent whose status initiated the conditional transfer in the first place (and all remaining tape constituent states will also be the same). Thus the ultimate effect of laying down special marks on the tape molecule and then removing them while we move through the main program molecule, is to transf activation and contact to any desired particular program constituent (determined by the location of the conditional transfer and the number of special marks its loop lays down). Since the program molecule is in the form of a continuous loop, a sufficiently large number of shifts in one direction along the program molecule will always transfer us to any desirec program constituent.

Clearly this structurally simplified system can carry out any of the computational actions of the earlier system, including universal computation; suitably augmented, as before, with synthesizing constituents and the like it can carry out any of the constructional including reproductive activities of the earlier systems.

II.4 Another Simplification

We now show how the structural form of the program molecule can be further simplified. In the simplified form of the last section we employed transfer loops appended to the circular main program; these appended loops imposed specific numbers of special marks on the tape molecule. By again slightly augmenting the properties of our individual molecule constituents, we can incorporate the transfer loop information into the circular structure of the main program.

This is done as follows. At each of the conditional transfer constituents we make the appended transfer loop of constituents part of the main circular program. If the transfer condition is *not* satisfied, then normal activation and contact is passed through the transfer region with no action taking place: the transfer region acts like a sequence of "no operation" constituents. At the end of the transfer region, the constituent activation and tape molecule contact will have its usual effect on ordinary main program constituents. If the transfer condition *is* satisfied, then the conditional transfer constituent will *specially* stimulate the first constituent of the transfer region now inserted in the main circular program structure. The constituents of the transfer region will, by a series of *move right* and *print special mark* actions lay down a superimposed sequence of special marks on successive tape constituents. At the terminal end of the transfer region, activation and tape contact is passed to the main program.

The main program constituents will (as in our last system) remove speciai m and move left on the tape. When all special marks have been removed, the main line program element newly activated will be in contact with the tape constituent whose status initiated the conditional transfer in the first place, and normal computation resumes.

It should be noted that in the course of a transfer operation with its shifting of activation through the main program to the intended transfe target constituent, additional, unactivated transfer regions will be encountered as well as "normal" computational regions. The transfer region const tuents contacting a special marked tape constituent, will *not* (in contrast to the ordinary computational constituents) carry out a "remove special mark and move left" action, but will instead merely transfer activation and contact to the next immediate program constituent neighbor. (If we did not impose this "ignore" property, a transfer region might have to be not only as long as the main program, but as long as the main program plus all transfer regions themselves, and attempts to implement this will produce an infinite regress.)

We conclude that (with some increase in complexity of individual constituent properties) "pure" circular form program molecule systems can carry out the same computational and constructional actions as our earlier systems, and are thus capable of universal computation as well as reproduc

II.5 A Further Simplification

It will be recalled that the circular form of the last two examples was employed so that a sufficiently large number of shifts of activation in the "normal" direction of the computation would transfer the point of activ tion anywhere in the program (including *earlier* points in the program). If we introduce the capability to shift action in *both* the normal and anti-normal direction in the program molecule, then the circular program form is no longer a necessity and a program molecule in the form of a single finitely long string of constituents will suffice.

There are several ways to implement this notion. Perhaps the most straightforward would be to have a transfer region, if activated, superimpose on successive tape molecule constituents a series of one of *two* kinds of special marks according as attention is to be shfited in the normal or anti-normal direction. Then, employing *mutatis mutandis*, the same techniques as in the last case, we can achieve the desired computational effect, and thus also the same constructional and reproductive properties will obtain.

Up to this point, we have always had to appeal to the notion of automatic self-assembly of program molecules (as a consequence of structural properties of connected individual constituents) in order to produce the required tertiary conformation. Since we have now shown that the program molecule can always take the form of a simple string of constituents, this strong self-assembly assumption is no longer necessary; construction of a program molecule can consist of the one-by-one synthesis and simple linking of program molecule constituents.

II.6 Broadcast Signalling Between Program and Tape

In a natural biological interpretation of our program molecule and tape molecule artificial molecular machine system, the active program molecule plays the role of the cell enzymatic machinery, while the tape molecule plays the role of the genetic nucleic acid passive informational molecule.

In this section of the report we will continue to revise and redesign our artificial molecular system in order to bring it into closer correspondence with what is believed to be the nature of real molecular genetic systems.

A major "unreality" of our present system has been the necessity (despite the program molecule structural simplification) for communication by direct sliding contact between the program molecule and the tape molecu

This necessity for direct sliding contact can be eliminated if we assume that for each program molecule (of whatever form we employ: arbitrary graphical, planar, single loop, or simple string segment) and for each tape molecule some additional properties are assigned to the individual constituents.

What we shall require of our basic program and tape constituents is the following:

1) at any moment of time at most one tape and one program constituent will be in an activated state.

2) An activated tape molecule constituent will continually "call its name" ("blank", "0", "1", "special mark", etc.).

3) An activated *CT* (conditional transfer) constituent will detect tape constituent name signals, and will act appropriately. That is, the CT constituent will, if the condition is satisfied, pass activation to the transfer target constituent (or to the apparatus implementing the transfer) and if not, will pass activation to the next immediate program constituent.

4) An activated *write* constituent (i.e. w(0), w(1), w(special) etc.) will send signals to change the status of the activated tape constituent.

5) An activated tape constituent will upon receipt of a *write* signal act to assume the signalled state (and once assumed, the activated tape constituent will begin to call its (possibly new) name).

6) Upon return receipt of the same sign as its command (confirmation o tape constituent status) a *write* constituent will pass activation to the *next* immediate program constituent.

7) An activated *move* constituent (R or L) will emit move signals, but when an activated tape constituent receives its first such signal the tape constituent will enter into an intermediate state where it sends a train of confirmation signals to the originating activated *move* constituent

When the move constituent receives this confirmation of recenpt it ceases the emission of move signals, then sends a "complete action" signal to the waiting tape constituent, then passes activation to the next program molecule. When the waiting tape constituent receives the order to complete its action, it passes activation to the called for tape constituent to the right or left.

(This complicated implementation of the *move* operations is designed to prevent *multiple* moves triggered by a single original train of move signals.)

8) Finally, activated *halt*, *sever*, *delete* etc. constituents will have their obvious effect on other activated constituents.

The above repertoire of actions will enable the two principal components of our artificial molecular machine system to communicate without the necessity of direct sliding contact throughout their lengths. (It should also be noted that this broadcast signalling tactic has assumed some "background" properties" quite natural to biologists, but not necessarily to automaton theorists. viz. the presence of an envelope housing our system, a fluid medium to support and transport constituents and signals, and prompt decay of emitted signals.)

II.7 Dispersed Program Constituents

Having introduced the notion that the two principal molecular components of our system can communicate with one another by broadcast chemical signals, it is plain that the same tactic can be employed to pass activation from one region of the program molecule to another, or even from one separate constituent to another, thus obviating much of the complicated implementation of conditional transfers. If separate program molecule constituents can "speak directly" to each other, then the program component structural form can be quite arbitrary, since it is no longer required to perform an activation communicating-routing function. In particular, there is no reas why the program constituents can not be separated and dispersed freely through the fluid medium, as long as we are willing to augment sufficientl our signalling transmission and receipt capability.

Adopting such a course would do much to bring our artificial system closer to actual molecular biological systems. Our dispersed but intercommunicating program constituents would have as their counterpart the dispersed protein-enzymatic machinery of the cytoplasm, while our string-li tape molecule has its counterpart in the DNA.

The inter-communication of our program constituents could be implement as follows: Each program constituent must be able to call the unique "name" of its immediate successor (conditional transfer constituents, must be able to call selectively the names of either of their two potential successors). A constituent will always be receptive to its name; upon receipt of its name, it will send an "acknowledgement" signal, which will cause the calling of the predecessor constituent to cease; activation is then assumed by the newly called constituent.

Note that we must have a number of distinct signals of the order of the length of the program (numbers of distinct program constituents). This is the price we pay for decomposing the structural communication connectio To the extent that we retain some of our original structural connections we can reduce the number of distinct signals required. It should also be noted that the notions of computational and constructional universality assure us that the size of omni-competent computers and constructors need not increase indefinitely; indeed there are known to be quite modestly sized programs which are nonetheless universal.

As to communication between a dispersed population of program constituents and the tape molecule, we can either employ the broadcast signallin already discussed in the last section or we can introduce the notion that an activated tape constituent "attracts" a freely moving activated program constituent, and that the fundamental computation is implemented by intimate rather than distant interaction between the constituents. This notion has more credence if we permit *multiple* copies of the dispersed program, operating in synchrony, so that the likelihood of activated constituents achieving the required proximity and orientation is increased.

II.8 Reproduction of Molecular Machine Systems with Dispersed Program Constituents

In Part I of this report we described various strategies by which artificial molecular machine systems might reproduce themselves. Those earlier systems consisted of two interacting chains of connected basic constituents. In the last section of this second part of the report we described an artificial molecular machine system consisting of a *dispersed* population of program constituents as well as a connected constituent string tape component, both housed in a fluid filled envelope and communicating, at least in part, by "broadcast" emission of chemical signals. The capabilities of our earlier described artificial molecular machines -- that they could compute, construct, describe themselves, and reproduce -- hold also for the systems described in the last section. The physical properties of our machine system have been so radically transformed though, that a complex process such as reproduction can no longer take precisely the same form as before.

For example, in our original system, the program machine consisted of an interconnected structure of individual program constituents and the *form* (and thus the routes of activation and communication) was a consequence of the assumption of "self-assembly": once the primary connected sequence of program constituents was produced the secondary and tertiary structure (with its pattern of required interconnections) followed automatically. Here, however, in our present program machine with its dispersed constituen communication is carried out between program constituents by means of "unique" emitted and received "addresses" or "names" of individual basic program constituents.

Thus, the larger the program machine (the more constituents of which i consists) the more different unique names or addresses will be required, and in particular, the more different emitters and receivers of the names or addresses. This leads, in a sense, to more *different* basic types of program constituents.

We thus have a situation in which an original fixed finite number of basic program constituents with an original fixed finite number of kinds of synthesis constituents and routines, must be capable of creating all basic constituents along with *unique* addresses not merely for each *type* of basic constituent, but for each *token* of each type which may be employed. This looming regress can be circumvented by any one of several tactics.

In the first approach (which we will employ later in our description of reproduction) we can assume that each basic program machine constituent type comes equipped with incompletely determined emitting and receiving organs or regions. The real problem will be to assign the proper settings each of those organs or regions so they receive and emit the proper require unique messages. This assignment can be made by having the tape molecule, from which the construction information is being read, contain not only the information as to the next type of basic constituent to be produced, but the proper settings for each constituent receiving and emitting organ.

A second way of solving the problem is to make the assumption that the basic program machine constituents specified in the tape machine sequence, are always to be addressed in some *systematic* fashion by the active program machine. The program machine being "cognizant" of the address assignment scheme employs it to set signalling values in the basic constituents one by one as they are produced. Now since in general the communication structure among program machine constituents can be arbitrarily complicated, (and we may wish to submit to the actively constructing program machine descriptions of *any* machine) the program machine may not know what addressing scheme to employ. If however the program machine "knows" that the machines to be constructed are to be (for example) based on the *simple string* form of II.5, then the addressing scheme is simple: the communication-activation relationships inside the program machine are always between next neighbors to the right or left. Thus if the constituents are described in order on the tape, the first constituent can be given address 1, the next 2, etc., and each could be "told" the addresses of the right and left neighbors with which it may have to communicate.

We now proceed to outline the process by which our dispersed constituent program machine artificial molecule system could reproduce itself.

Since our system as presently conceived, contains almost no means by which constituents can be physically sorted, ordered, segregated, etc. the form of reproduction described, through satisfying essential reproduction conditions, will be extremely rudimentary. Let us picture our systems of program and tape machines residing in fluid droplets on a flat surface. External environmental conditions will periodically redistribute, disperse, augment and diminish the number and size of these droplets. We begin with an active program machine (in dispersed form) and a passive tape machine (in the form of a connected string of constituents) residing in a single droplet.

The tape will contain a complete blueprint for the construction of a program machine. The parent active program machine initiates the reproductive cycle by commencing to read and act upon the tape contents. (See II.6 for the manner in which the reading is done.) As in the construction process described in Part I, the active program will, once having ascertain from the tape the nature of the first constituent to be produced, embark on a synthesis process, creating or recruiting the new basic constituent required, setting its broadcast chemical receiving and transmitting "organs" to the proper values. Continuing this process the program machine reads the tape and from the description or instructions there produces a population of program constituents. If the described population is identic to that of the parent program molecule itself, then duplication of the pai program machine has taken place. Note also that the parent program machine could readily have produced *numerous duplicate* copies of the program machin constituents.

The next action of the active program machine is to make one or more copies of the tape machine itself. Having completed this task, essential reproduction has taken place: we began with a system consisting of a singl tape machine and a single program machine, and we end with two or more copies of the original pair of machines.

Since our formal system is "underfurnished" with means actively and autonomously to transport componentry, locate it precisely, and segregate it, we shall have to invoke random environmental assistance in dispersing and resegregating our basic artificial machine components. Our initial droplet of fluid may be combined with other droplets and dispersed and recombined. If a droplet contains a full complement of active program constituents, and a tape machine, then the reproductive process can contin

Survivability of such systems, so subject to environmental vagaries is not very high. There is no logical bar, however, in anything we have proposed for our artificial systems, to the program machine creating membranes, a mitotic apparatus, etc. and all this additional capability being contained in the original tape description and synthesized into the capabilities of an active program machine. Such augmentation would of course greatly enhance survivability. The essential logical features are already present by which a continued intricate adaptive and evolutionary course of action can take root.

II.9 Evolution for Artificial Molecular Machine Systems

In the final remarks of the last section we suggested that our rudimentary artificial molecular machine systems already possess all the essential *logical* properties required in order to exhibit evolution from simpler to more complex forms of behavior. This is so because our machines possess universal computational powers as well as constructional powers, and Myhill [1964] has shown that such machines can exhibit evolutionary behavior. The essence of his argument described for our machines is this:

An artificial molecular machine system possessing universal computational powers (as well as constructional powers) can instantiate any formal axiom system. That is, given the set of axioms and rules of inference of the axiom system an artificial molecular machine can act so as to generate all and only the theorems of the formal system; given any putative theorems and a list of expressions in the formal system which purports to be a proof of the supposed theorem, this claim can be checked by the molecular machine system, etc.

Such capability of an artificial machine system can be seen to have survivability value, for, under interpretation, a formal system could be seen as explicating relationships and *consequences* among matters of vital importance for the success of the organism. (Any organism which can deduce by calculation that continued actions of a certain sort will cause it to "fall off a cliff", is more likely to survive than an organism which cannot calculate such consequences.)

Godel's famous incompleteness result [Godel 1931] tells however that once certain axiom systems attain a certain expressive power they will also possess certain inherent fundamental limitations: there will always be truths expressable in the system but not provable as theorems.

For the machine which is organized to be an instance of such a formal system, these unprovable statements can be located, and an arbitrary decision made to hereafter take the statement or its negation as an axiom. The original machine can now undertake to modify itself (learn or adapt) or can *construct* an offspring machine embodying the new axiom system. In a separate paper, Godel [1936] has shown that in the resulting augmented new formal system, not only can we always prove more theorems, we can prove many of the old theorems faster. Thus the modified machine or the offsprin machine could not only surpass its parent in deducing additional consequence about its static environment or exploring alternative futures under various assumptions and choices, it might get some answers faster than the parent (a property frequently exhibited by children) and also very possibly a survivability enchancing property.

Of course, for this new internalized formal system there will again be statements which can be neither proved or disproved. The arbitrary inclusio of this new statement or its negation as one of the axioms can again be undertaken, with similar consequences: our artificial organism will be able to do more, faster than before. Continuing in this fashion, employing only deterministic tactics, we can produce a sort of indefinitely continuable improved performance, either in a single machine undergoing "learning" or "adapting", or in a series of evolving machines.

II.10 Essential Logical Properties of General Self-Duplication

Although we have presented only a few of the possible variations of molecular machine self-reproduction, let us abstract from those examples some persistent (and perhaps essential) features of the logical and information transactional process of creating self copies.

First, the fundamental reproductive machinery must always exist in at least two copies though the precise initial line of division of information between active (protein) and passive (nucleic acid) components may be shifted). Use of two copies means we can avoid any (potentially) disabling paradox of self-reference, for ultimately the process is carried out by a device *not* examining and acting upon itself.but by acting upon a *copy* of itself, or a description of itself, or a set of instructions for constructing itself.

It thus also becames clear that the frequently heard remark that DNA replication is by itself an instance of self-reproduction is very misleading. For DNA by itself cannot replicate; an active environment enzymatic cytoplasm machine must exist to actin conjunction with the DNA; both the passive DNA machine as well as the active enzyme machine must be copied before complete self-reproduction has taken place. Thus, known biological self-reproduction minimally entails both (at least some) DNA replication and (at least some) protein synthesis.

Secondly, the two (or more) copies of the same information can differ greatly in form. Although each copy will contain (at least some of) the same information, the coding systems may be very different, and the physical vehicle of the separate copies of the information also may differ radically.

Third, the copies may be said to play active or passive roles, but this is likely to be in part as much a product of our way of looking at processes and describing the processes, as any "essential" aspect of the self-reproducing system *per se.* It is, for example, convenient to *speak* of one copy "reading" or acting upon another; but it may be only that: a way of speaking that follows an expositional convenience or convention, the two copies actually interacting or cooperating to carry out the reproductive process.

Fourth, carrying out self-reproductive processes of the generality desired requires that the system must have obtained some fundamental degree of complexity. In particular the "controller" must possess some essential reading and acting powers.

We have expressed these essential reading and acting powers as specifi properties of an active program machine (which plays the role of the enzyma environment of a passive description machine). It may be argued that a "simple templating" from the passive description machine (which plays the role of DNA) accomplishes the same tasks and is both more "natural" and less complicated than the "reading and acting upon an abstract description' mode of construction generally espoused by automaton theorists. In "simple templating" though, the environment implicitly possesses a considerable nu of active properties. For a genuine comparison to be made between the "natural" templating process and the automaton process of construction the properties of the whole "template plus environment machine" should be made explicit.

Indeed an examination of templating and of automaton reading and acti reveals that they are not essentially different for, at bottom they both satisfy the same distinguishable requirements. For example in both systems there must somewhere be a designated accessible location which cont the information on the *type* of constituent presently required, a designated accessible location which contains the information as to the *site* where the presently required constituent is to be attached, a place and means of recruiting or manufacturing a constituent of the required type and transporting it to the proper site, and a way to abandon one task and shift to the next.

The separate and distinct nature of these (and other) tasks is in templating, often obscured since from a gross point of view all the actions, sites and locations, coincide. The distinguishing and naming of the actions which the automaton theorist deems necessary has the virtue of bringing to the attention and clarifying what seems to be the minimal logical essential steps in the process and forcing us to examine the "micro-structure" of the templating process to find the equivalent distinct actions.

II.11 Complexity of Self-Reproduction

Whatever precise tactics of reproduction are employed in biological organisms, we shall be most interested in those stratagems which are capable of supporting biologically necessary or important processes, such as evolution. In the system described here a largely passive stable genetic record exists which can be acted upon and transcribed for a new generation by an active genetic component. If we permit fortuitous changes in the passive record, these changes can, by means of the active component be copied and carried forward as the *passive* record of the next generation while also being embodied and tested for fitness in the *active* componentry of that generation.

This simple essential core of reading and copying powers carried in an interacting active and passive form prompts the following question: computationally, just how potent is this minimum adequate genetic apparatus? In this paper we have shown that our artificial molecular machines are capable of carrying out the computation of *any* Turing machine. As

Minsky and Thatcher have pointed out however (in Lee [1963]) such universal computational power is not in fact required of machines capable of sel description (or self-reproduction). Self-reproducing strategies, even of our "universal construction" type (where the active "parent" machine may be given a description of *any* machine constructable in the system and will produce that described machine as offspring) require only some simple copying and interpreting-constructing capabilities. Indeed, as we have seen, the basic active parent machine capabilities are as follows: presented with a passive machine description, the active parent will 1) copy the passive description, producing a second passive description machin (this is the analogue of DNA replication), 2) copy the passive machine description a second time, producing an *active* machine (this is the analog of protein synthesis), 3) conjoin the newly produced active and passive machines to form a new complete molecular organism.

Given this basic operational repertoire of a "parent" machine let us consider briefly some of what is implied by such properties and in particular what the immediate offspring of such a parent machine may be capable of.

Let us contemplate what happens when (as with a Universal Turing Machine) we consider submitting all possible (passive machine) description to a parent genetic machine. Clearly any desired passive machine and acti machine pair can thereby be produced (for example the parent, from the appropriate description, can construct a machine which can carry out any Turing computation including of course the computation of a Universal Turing machine). By the normal genetic process of our parent machine, the new active machine and its passive copy will be conjoined. Of course if we wish to carry out particular computations we shall need a way of submit to any newly produced active machine (including if we wish, an active Universal Turing Machine) a particular passive machine *problem* of our choosing, not solely the passive description of our newly produced active machine itself.

We can get around this difficulty as follows. We can submit to an active parent genetic machine a passive description of a machine which possesses the desired computational properties, *and* which has appended to its description an additional segment which describes the *problem* the desired machine is to work on. Now of course when the parent machine produces the new active machine, this new machine will possess some superfluous portion, a consequence of the appended problem. Similarly when the parent machine produces the new *passive* machine, there will be, in addition to the desired problem description, a superfluous portion describing all of the newly produced active machine. Having the proper foresight however, we can always design our new active machines so that in addition to having the desired computing powers they will never enter into the superfluous portion of their componentry (the portion resulting from the *problem* appended to their description) and will ignore, in their computational task, the portion of the passive description presented to them which is merely their own description.

We therefore have the following result. A "parent" genetic machine though not itself capable of directly carrying out an arbitrary Turing computation, can construct active offspring machines capable of carrying out any Turing computation; thus, in the context of successive generations of machines from descriptions, the simple genetic operations do imply systems possessing universal computational powers.

II.12 Final Remarks

In this report we have made an attempt to achieve a rapprochement between automaton theoretic approaches to reproduction and contemporary biological conceptions of the genetic reproductive process.

It is hoped that the artificial machine approach to biological theory espoused here will entice automata theorists to attempt close explications of empirical phenomena, and will thereby yield numerous fruitful, revealing and rigorous explanations which can serve as genuinely competitive alternat empirical hypotheses.

The possible specific future directions and emphases are many. For example, one natural continuation of this research program would be to expl cate in automaton theory terms, the growth of cell membrane and cell wall, and from thence the development of integrated multi-cellular organisms. Another direction might be to refine the automaton terms of the discussion, giving a more detailed and mathematically precise description of all the processes assumed. We might also wish to broaden the scope of the model to in corporate automaton counterparts of additional known cellular and sub-ce ular events. Finally we might attempt increasingly to "empiricize" our automaton theory primitive notions to reflect more minutely the latest and most adequate conceptions of physical, chemical, and biological reality

REFERENCES

BERNAL, J. (1965). *Theoretical and Mathematical Biology* (T.H. Waterman and H.J. Morowitz, Eds.), Ginn, Boston, 96.

GODEL, K. (1931). "Uber formal unentscheidbare Satze der Principia Mathematica und verwandte Systeme I." *Monatshefte fur Mathematik und Physik, 38,* 173.

GODEL, K. (1936). "On the length of Proofs". *The Undecidable,* ed. M. Davis, Raven Press, 1965, 82. (Translation from German).

HOLLAND, J. (1974). *Adaptation in Natural and Artificial Systems.* University of Michigan Press, Chpt. 8.

LAING, R. (1973). *J. of Cyber. 3,* 2, 16.

LEE, C. (1963). *Mathematical Theory of Automata.* Polytechnic Press, Brooklyn, 155.

LOFGREN, L. (1972). *Trends in General Systems Theory.* (G. Klir, Ed.) Wiley-Interscience, 346.

MYHILL, J. (1964). "The Abstract Theory of Self-Reproduction", *Views on General Systems Theory,* ed. M. Mesarovic, J. Wiley and Sons, New York, 106. Also: *Essays on Cellular Automata,* ed. A.W. Burks, U. of Illinois, 1970, 206.

ROGERS, H. (1967). *Theory of Recursive Functions and Effective Comoutabilitu.* McGraw-Hill, 188.

SMITH, A. (1969). *Cellular Automata Theory,* SU-SEL-70-016, Stanford Electronics Laboratories.

THATCHER, J. (1963). *Mathematical Theory of Automata.* Polytechnic Press, Brooklyn, 165.

THATCHER, J. (1970). *Essays in Cellular Automata.* (A.W. Burks, Ed.) University of Illinois Press, 103.

TURING, A. (1936). *Proc. London Math. Soc., Ser. 2, 42,* 230.

VON NEUMANN, J. (1966). *Theory of Self-Reproducing Automata.* (A.W. Burks, Ed.) University of Illinois Press.

VON NEUMANN, J. (1951). *Cerebral Mechanisms in Behavior: The Hixon Symposium.* (L.H. Jeffries, Ed.) John Wiley, New York, 1.

WAGNER, E. (1967). *IEEE Conf. Record: 1967 Eighth Annual Symposium on Switching and Automata Theory,* 45.

WANG, H. (1957). *J. Assoc. Computing Mach.,* Vol. 4, p. 63.