Effective Processes and Natural Law

Robert Rosen

1. Introduction

One of the most remarkable confluences of ideas in modern scientific history occurred in the few short years between the publication of Gödel's original papers on formal undecidability in 1931, and the work of McCulloch and Pitts on neural networks, which appeared in 1943. During these twelve years, fundamental interrelationships were established between logic, mathematics, the theory of the brain, and the possibilities of digital computation, which still literally takes one's breath away to contemplate in their full scope. It was believed at that time, and still is today, over half a century later, that these ideas presage a revolution as fundamental as that achieved by Newton three centuries earlier.

The name of Alan Turing is preeminent in the history of these astonishing developments. For it was Turing, in his seminal paper of 1936, who first really juxtaposed the relevant ideas through the construction of the class of "machines" which bear his name. These *Turing machines* were, on the one hand, explicitly extrapolated from the mental processes of a human being engaged in a mathematical computation; on another hand, they represented a formal embodiment of logical or algorithmic processes as manifested in mathematics; and on yet another hand, through the use of the term "machine" they suggested both the harnessing of material processes to extend our own mathematical capabilities (soon to be realized through the creation of digital computers) and, at an even deeper level, a new and powerful metaphor for exploring life itself.

At a purely mathematical/logical level it was quickly recognized that the Turing machines were one of a number of equivalent formalisms for embodying the concept of an *algorithm.* In its turn, an algorithm is regarded as the epitome of an *effective process* for solving a problem. Now an effective process connotes an idea of absolute necessity; it is an inferential chain which in every case *must* lead from appropriate initial data to the corresponding answer or solution. Moreover, an algorithm is a *rote* *process* which, once set in motion, requires no further intervention, no reflection, and no thought in its relentless progression from data to solution. That is why it seems, in retrospect, so natural to embody it in a "machine", as Turing did.

Since the notion of "effective process" is an informal, intuitive concept, while the notion of "algorithm" is a precisely formalized mathematical one, it was early suggested that the latter can replace the former. This is precisely the substance of Church's Thesis (cf. Kleene *1952*), which asserts that any process one would want to call "effective" can already be carried out by some properly programmed Turing machine.

Now, strictly speaking, all of the developments described so far are entirely formal; they take place in a logical and mathematical universe of propositions and production rules. In a sense, they are "all software". However, much of their interest resides in the fact that terms like "machine" or "effective" have connotations which are nonmathematical, which pertain to the external world of natural, material phenomena. Indeed, as we have already noted, Turing himself designed his mathematical machines as abstractions from a real-world phenomenon, a human being performing a computation. The suggestion is thus irresistibly conveyed that if *this* aspect of human mental activity can be "mechanized", why not others? Why not all? Likewise, if mental processes, involving what happens in a *material* brain (i.e., in hardware) can be represented entirely formally, why can there not be other kinds of material systems (i.e., other hardware) which can be made to do the same thing the brain does? Here we see, in embryo, the field of "artificial intelligence" in its widest ramifications, and much else besides.

But once we have admitted a material significance to words like "machine" or "effective", we have left the world of mathematics and entered the world of (in the broadest sense) physics. And whereas the Turing machines are, as we have noted, "all software", physics is by contrast "all hardware". We have thus, for good and ill, introduced a fundamental distinction between hardware and software which is, in itself, not part of the formal theory with which we began; nor, by the same token, is it part of physics either. This distinction, as we shall see, is vital, but it is insidious; it is camouflaged in the all-encompassing, umbrella term "machine". We can see how insidiously the distinction creeps in from the following quotation taken from Martin Davis's book *Computability and Unsolvability* (*1958*):

> For how can we ever exclude the possibility of our being presented someday (perhaps by some extraterrestrial visitor) with a (perhaps extremely complex) device or "oracle" that "computes" a noncomputable function?

This was clearly meant to be the most rhetorical of rhetorical questions, presented in the context of an entirely formal mathematical development. But we see here clearly the equivocation on the term "machine", resting on a tacit distinction between *real* hardware and logical software.

The present author has been troubled for a long time by the deep epistemological ramifications of this question. In particular, once we admit "hardware", or material systems, into our discussion (and as we have already noted, that was from the beginning the clear if tacit intention), what happens to the idea of an "effective process"? A long time ago (Rosen *1962*) I considered the question of what Church's Thesis means in this new context. Specifically: Is Church's Thesis a fundamental restriction on material nature (akin to the exclusion of the *perpetuum mobile* by the Laws of Thermodynamics), or not? And what happens to recursiveness if Church's Thesis can be violated by natural processes?

It may be useful to review the salient points of our earlier argument here. Suppose we are given a *physical* system S (perhaps the alien "computer" of Davis). Its behavior is governed by physical laws, which we learn about by doing experiments. A typical experiment will involve either doing something to the system (i.e, perturbing it from outside) or else letting the system do something to its environment, and then observing or measuring the result. Clearly, both the experimenter's intervention and the measured results are material events. Events are described or characterized by means of numbers, whose values are determined by the application of suitable meters (cf. Rosen *1978*). Suppose for simplicity that our experimental intervention α is characterized by such a number $r(\alpha)$, and that the resultant behavior of our system is characterized by another such number β. In this way, our experimenter can generate a table of values

$$r(\alpha) \longmapsto \beta$$

which defines a function f from numbers to numbers. The reader will recognize this procedure as a typical input-output characterization of our system S. The form of the function f clearly tells us something about the *laws* which govern the behavior of S. This is, after all, the whole function of experiment in science.

Now surely this experimentation process is in some sense *effective.* Indeed, as we shall argue at great length below, sequences of events in the material world (e.g., in the system S) are governed by *causal relations,* which bind them together quite as inexorably as implication relations bind propositions. Thus, if our experimental procedures are *repeatable* (which means that the same causal sequence in S can be recreated at will), then Church's Thesis must mean that *any input-output function f, generated as we have described from any material system S, must also be recursive or computable.* Otherwise, the system S would be precisely the "computer" which Davis assured us is excluded from possibility.

Seen in this light, Church's Thesis is an attempt to draw inferences or conclusions about hardware (physics) from premises about software (algorithms). Another well-known attempt to do the same thing is embodied in von Neumann's arguments about "self-reproduction" (Burks *1966*; Arbib, this volume; cf. Section 5 below). Here again, the intent is to learn something about the behavior of *material* systems (especially organisms) from a *formal* theory of computation. As might be expected, such enterprises are risky in the extreme; but *if* it could be successfully pulled off, the rewards would be great indeed.

At the very least, we can perhaps already see that introducing an idea of "hardware" into formal theory will have some peculiar ramifications. Complementary peculiarities arise from the other side, when we attempt to introduce ideas of "software" into physics. However, these very peculiarities promise to tell us something interesting about both. In the remainder of the present paper, we shall explore some of these possibilities.

2. *Church's Thesis in Formal Systems*

The essence of Church's Thesis is that it identifies *logical inference* in any formal system with *string processing*. In its turn, string processing, or word processing, is a purely *syntactic* activity. String processing is, of course, precisely what the Turing machines do. Nevertheless, it seems on the face of it to be a very strong, perhaps excessively strong condition to require that every inference in a formal system should be expressible in syntactic terms *alone*; i.e., that every trajectory from premises to conclusion should be navigated entirely through the manipulation of the symbols in which these propositions are encoded.

Nevertheless, a rather strong case can be built to support this rather unlikely-looking Thesis. It distills a trend towards formalization which began with Euclid, became a matter of urgency in the confusion following the discovery of non-Euclidean geometries (i.e., geometries which, as formal systems, were consistent as God-given Euclid), and of absolute desperation when the paradoxes in naive set theory were revealed. The formalistic response to this situation, pioneered by David Hilbert, was precisely to empty mathematics of any semantic content whatsoever, arguing in effect that it was a needless semantics which was at the root of the difficulties. This in effect turned all of mathematics into a kind of game in which meaningless symbols were manipulated according to (a finite family of) arbitrary syntactical rules. Indeed, the whole point of Hilbertian Formalism is to create systems in which there is nothing but syntax.

Perhaps the clearest statement of this kind of Formalist program was given by Kleene *1952*:

> This step (axiomatization) will not be finished until all the properties of the undefined or technical terms of the theory which matter for the deduction of theorems have been expressed by axioms. Then it should be possible to perform the deductions treating the technical terms as words in themselves without meaning. For to say that they have meanings necessary to the deduction of the theorems, other than what they derive from the axioms which govern them, amounts to saying that not all of their properties which matter

> for the deductions have been expressed by axioms. When the meanings of the technical terms are thus left out of account, we have arrived at the standpoint of formal axiomatics ... Since we have abstracted entirely from the content matter, leaving only the form, we say that the original theory has been *formalized.* In this structure, the theory is no longer a system of meaningful propositions, but one of sentences as sequences of words, which are in turn sequences of letters. We say by reference to the form alone which combinations of words are sentences, which sentences are axioms, and which sentences follow as immediate consequences of others.

Clearly, the idea here is that it is always possible to replace *semantics* ("meanings") with syntactics, so that *logically* no information is lost; any inference involving semantics possesses a purely syntactical image in the formalization.

In such a formal system, we start with the idea that the axioms are *true.* This notion of truth is *hereditary*; if the axioms are true, then so also are the symbol sequences obtained by applying the inferential rules of the system to them; thus truth passes from axioms to theorems. So far, we never need to import a notion of "truth" into the system from outside, as it were; we simply construct true propositions (theorems) as we go along.

The troubles embodied in Gödel's celebrated theorems (Gödel *1931*) arise from trying to compare this constructive notion of internal truth with preassigned *external* truth-value in formal arithmetic; as Gödel showed, they do not match. Furthermore, one way of looking at Turing's theorem on decidability (Turing *1936-7*) is that there is *no internal inferential mechanism* for deciding whether a proposition in the given formalization is a theorem (i.e., true) or not.

Thus, we *know* that the Formalist program, in which only purely syntactic inferences are allowed, is too impoverished to even play the game of number theory. That is, we must either allow into our system some "informal" inferential procedures, which according to Gödel's Theorem *cannot* be reduced to syntactics even in principle, or else restrict ourselves forever to mere fragments of number theory. Such "informal" inferential procedures are what are disallowed by Church's Thesis; they are *ineffective.*

Let us put the above discussion into more familiar terms. In ordinary ("Platonic") mathematics, which of course has both a syntactic and a semantic aspect, we know that if we are given a set, a variety of other sets can always be built from S via canonical constructions. For example, we have the power set 2^S; we have the free algebraic structures (semigroup, group, etc.) generated by S, we have the set $H(S, S)$ of all maps from S to S; we have the Cartesian product $S \times S$, etc. These associated sets give us *inferential capabilities* which have the power of logical relations, without necessarily being expressible in terms of the formal inferential laws which govern the mathematical system from which S came. For instance, if $Q : S \to S$ is an automorphism of S (i.e., an element of $H(S, S)$), and $s \in S$, we may say that $Q(s) = s'$ establishes an implication relation between s and s'. But this "implication" need not coincide with any we can draw from the production rules governing the system; i.e., need not follow from system syntactics alone. If it does, we may say that our Q is *computable* in the system; otherwise, *not computable.* In the latter case, we would have to say that the mapping Q is not *effective,* according to Church's Thesis. But clearly, if Q has any meaning or existence at all, once it is given to us, it *is* effective.

In the "all-software" world of formal systems, we can of course restrict ourselves in any way we like. Thus, we can agree not to allow ourselves any automorphisms Q which cannot be expressed in purely syntactical terms. In such a world, and only in such a world, could Church's Thesis hold unrestrictedly. Whether such a formal world would be at all interesting is, of course, another question. And as we shall see, the situation gets even worse when we allow "hardware" into our world.

Before turning to material systems, we should say a word about the encoding of propositions in a formal system onto Turing machine tapes, and decoding tapes back into propositions in the system. It is fairly clear that we may use the word "effective" in its usual intuitive sense in connection with encoding and decoding; the familiar Gödel numbering, for example, is clearly an effective mapping from syntax to arithmetic and back. Indeed, in showing that a process in some formal system is effective, or recursive, we can clearly combine the encoding, the computation, and the decoding into a single Turing machine which does all three. However, we merely note here for future reference that the encoding and decoding are logically distinct from each other and from the actual computation; if these are in some sense "ineffective", then Church's Thesis may appear to fail, even though the computation itself is completely recursive.

3. *Implication and Causality*

As we have seen, in the realm of formal systems, Church's Thesis identifies the intuitive notion of "effective process" with the purely syntactic idea of string processing. That is, any implication which can be "effectively" performed within the system can already be carried out by means of a finite set of production rules, operating on finite strings of symbols taken from a finite alphabet.

We have also pointed out that when we deal with the *material* world (as opposed to the formal ones of mathematics and logic) ideas of implication are replaced by ideas of causality. Nevertheless, we can still retain the idea of an "effective" process. In this context, Church's Thesis means that any *causal* sequence can be represented by a corresponding recursive process; i.e., *any causal sequence can be described by purely syntactic means.* If this is true, it of course places severe limitations on what *physics* can be like. *The question is no less than whether the Laws of Nature can* *themselves be formulated in purely syntactical terms, or whether they can possess an inherent semantic component which cannot be finitistically formalized.*

It should be noted that the urge to formalization (i.e., to pure syntactics) in mathematics is exactly parallel to similar trends in theoretical science. Indeed, the whole thrust of atomic theory (or nowadays, the theory of "elementary particles") is to reduce all material processes to the motion of ultimate constituent units, devoid of any internal structure ("meaning"), possessing only an instantaneous position ("configuration") and the temporal derivatives of position. The forces which push these ultimate units around are the precise analogs of the production rules in a formal system. Hence the paths or trajectories traced out by a material system under the influence of given forces are the analogs of formal theorems, with initial conditions as axioms.

The idea that causal relations between events in material systems can be related to implication relations between propositions describing those events is the *sine qua non* of theoretical science. Indeed, the belief in what used to be called *Natural Law* requires (a) that the sequences of events we perceive in the external world are not arbitrary or whimsical, but are governed by definite rules (this is *Causality*), and (b) that these rules can be articulated in such a way that they can be grasped by the human mind. Taken together, this formulation of Natural Law asserts precisely that causal relations in material systems can be brought into congruence with implications in a formal (ultimately, mathematical) system of propositions about those events.

This situation can be most succinctly expressed in terms of a diagram (see Figure 1).

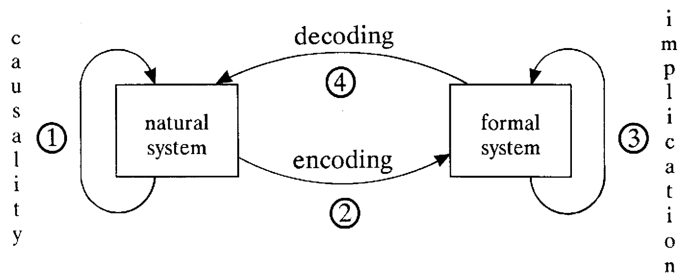

Figure 1.

We say that a *modeling relation* exists between the natural system on the left of the diagram, and the formal system on the right, when the following commutativity holds:

$$① = ② + ③ + ④ \tag{1}$$

That is, we get the same answer whether we simply sit as observers, and watch the unfolding sequence of events in the natural system, or whether we (a) encode some properties of the natural system into the formalism, (b) use the implicative structure of the formal system to derive theorems, and then (c) decode these theorems into propositions (*predictions*) about the natural system itself. When the diagram commutes, we have established a congruence between (some of) the causal features of the natural system and the implicative structure of the formal system. We can then say that the formal system is a *model* of the natural one, or alternatively, that the natural system is a *realization* of the formal one.

These little diagrams themselves possess a number of rich and important epistemological properties, which we cannot enter into here; for a fuller discussion, see e.g. Rosen *1985.*

Once we have thus constructed a formal system which is a model for some natural process, we have left the realm of science and entered that of mathematics. We can then treat a model as we would any other formal system. In particular, we can look at its purely syntactic aspects, which we can immediately identify with the "effective" processes of Church's Thesis, and ask whether these exhaust the implicative resources of the system itself.

In this way, we can construct a purely syntactic "machine" model of our original natural system, as indicated in Figure 2.

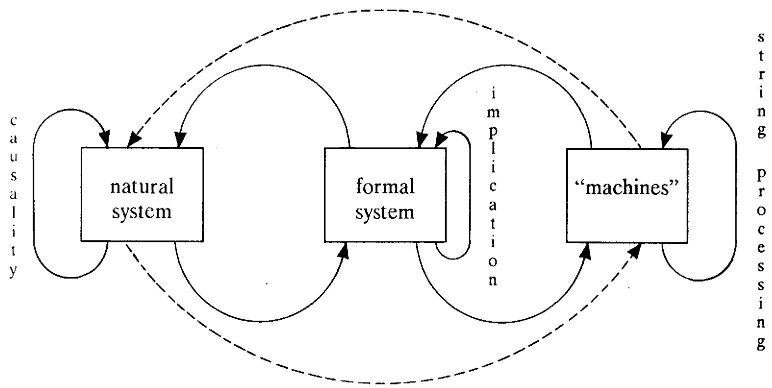

Figure 2.

This can be seen by looking only at the outer two systems, and forgetting about our original model, which now plays the role of a "transducer" between them.

However, when we do this, the following points must be explicitly noticed: (a) the *encoding* and *decoding* arrows between them (the dotted arrows in Figure 2) cannot be described as *effective* in any formal sense, and (b) these encoding and decoding arrows involve exclusively the *input* and *output* strings of the machines inhabiting the right-most box. Both of these observations are important. We shall briefly discuss them each in turn.

In applying Church's Thesis to formal systems, we noted above that the corresponding encoding and decoding arrows themselves represented *formal* processes, which could in fact be amalgamated into the Thesis itself. However, when we wish to compare a natural system, governed by causality, with a formal system, governed by implication, this is no longer the case. The encoding instruments, or transducers, are now themselves material systems; i.e., governed by causality and not by implications. As noted above, they are (in the broadest sense) *meters.* Since these meters are governed by causality, anything they do is *effective* in a material sense. But it is clear that a formalization of the encoding process would require more *models*; these would in turn require their own encoding and decoding processes, which would require more models, etc., an infinite regress. It is precisely this fact which makes "the measurement problem" so hard in physics. The question as to whether this potential infinite regress can be terminated at some finite point is a deep epistemological question about the nature of the world, with close ties to such things as reductionism. We cannot of course enter into such matters here; we will merely assume that a set of meters or other transducers from the natural world to input tapes is *given,* and that Church's Thesis can be investigated relative to *these* encodings (which are then, by hypothesis, effective in the material sense).

Our second observation is also epistemologically important. It says that all relevant features of a material system, and of the model into which it was originally encoded (cf. Figure 1), are to be expressed as *input strings* to be processed by a machine *whose structure itself encodes nothing.* That is to say, the rules governing the operation of these machines, and hence the entire inferential structure of the string-processing system themselves, have no relation at all to the material system being encoded. The *only* requirement is that the requisite commutativity hold, as expressed in Section 1 above, between the encoding on input strings and the decoding of the resultant output strings. As we shall see, this is the essence of *simulation.*

We have already noted that Church's Thesis amounts to asserting that all causal relationships can be expressed in purely syntactic terms. We can formulate the Thesis still more sharply now: relative to any given encoding of a natural system into input strings, the string-processing machinery itself must not encode any aspect of the material system. That is to say, the "hardware" of the machines *must* be totally independent of the "hardware" generating the strings to be processed. If this cannot be done, then Church's Thesis cannot be true.

Given the above, Church's Thesis asserts that the Turing machines constitute a class of *universal simulators* for all material processes. There are then two questions to be asked: (a) is it true? and (b) if so, what does it mean?

4. *Is Church's Thesis Physically True?*

The upshot of the argument of the preceding sections is the following. Nature provides us with a plethora of material processes which we would want to call "effective". These processes are (we suppose) governed by causality, and not by implication or production rules, as in a formal system. In these terms, Church's Thesis can be expressed as follows. Given any such process, we can encode appropriate propositions about the natural system generating it onto a set of input tapes to a Turing machine; and the corresponding output tapes can be decoded so as to perfectly *simulate* the process in question. Equivalently, Church's Thesis asserts that all "information" about material processes, and hence all of Natural Law, can be expressed in purely syntactic terms.

We know from Gödel's Theorem that sufficiently rich *formal* systems always contain inferences which cannot be obtained syntactically. More specifically, given any encoding of propositions of the system onto input tapes to Turing machines, there will be propositions of the system which are "true" but will never appear on any output tape. The processes by which the "truth" of such inferences are established are thus *ineffective*; they cannot in principle be simulated by any Turing machine in the given encoding. We can always change the encoding, of course, but this will not change Gödel's conclusion.

Thus, in formal systems, we already find that a purely syntactical encoding will in some sense *lose information.* The information lost must then pertain to an irreducible, unformalizable *semantic component* in the original inferential structure. By changing the encodings, we can shift to some extent where this semantic information resides, but we cannot eliminate it.

By itself, this result of Gödel does not bear on the *physical* truth of Church's Thesis, since it is a purely formal result. But it is in fact suggestive of how the physical form of Church's Thesis might be verified or falsified.

As we have seen above, the manner in which we compare material processes with formal ones is through the establishment of modeling relations, as diagrammed in Figure 1 above. Formal models of material systems are then perfectly good formal systems, whose inferential structures by definition reflect causal processes in the natural system being modeled. Thus, if a *model,* arising in this fashion, should fall within the purview of Gödel's argument, this would at least be strong evidence that Church's Thesis is false as a physical proposition. Stated another way, there would exist physical processes which could effectively compute nonrecursive functions. It would also mean that Natural Law cannot be expressed entirely in syntactical terms.

The obvious thing to look for, then, is a model of a material system which is rich enough as a formalism to "do arithmetic". The formalisms which physics provides, as models of purely physical systems, are unfortunately extremely impoverished, considered simply as formal systems. But, as we have argued elsewhere, these formalisms are in fact highly nongeneric and do not suffice to image material systems like organisms (cf. Rosen, in press). One way of expressing this nongenericity is precisely in terms of the way they image causal structures. When this nongenericity is lifted, a new class of (*potential*) models is obtained in which the image of causal structure is infinitely richer and more complicated. Since it is precisely the formal imaging of causal structures which is at the heart of Church's Thesis, we may perhaps find in these formalisms many processes which are *causally* effective, but *mathematically* ineffective. This would mean that the behavior of such systems must contain an irreducible semantic component, one intimately related to the *complexity* of the system.

A different approach was taken long ago by John Myhill (cf. Myhill *1966*). He was able to show that, modulo some idealizations regarding measurement and performance tolerances, there are already classical analog devices (analog computers) which could "compute" nonrecursive functions. Such systems would thus already manifest behaviors which could not be predicted by any purely syntactic encoding, and hence would also have an irreducible semantic aspect.

Finally, we have already mentioned that Newtonian particle mechanics, and more recently, the unified physical theories based on elementary particles, are in themselves an attempt to express the Laws of Nature in purely syntactic terms. Insofar as any material system is comprised of such "meaningless" (i.e., structureless) elementary subunits, these theories at heart assert that to understand any behavior of any such system it suffices to describe it in terms of these subunits and their interactions. As noted earlier, this is the essence of reductionism.

In these terms, the familiar Laplacian Spirit is a purely syntactical concept: the embodiment of Church's Thesis if he could exist. As a matter of fact, he could not exist (or at any rate, not as a material system built of particles himself), for reasons we have already indicated. But even if he could exist, he would be a very poor biologist, for example; organisms, and open systems in general, are constantly turning over their constituent particles. Thus, to even find an organism, let alone follow it in time, he would need to supplement his purely syntactical information with other (semantic) information not formalizable within his system.

Thus, for a variety of reasons, there is cause to believe that Church's Thesis fails as a physical proposition. Nevertheless, as we have seen, to state and analyze the Thesis in material terms touches on some of the deepest and most basic aspects of theoretical science.

5. *The Role of Simulations*

We have already alluded above to the use of the Turing machines both as metaphors for the material world and as effective descriptions of that world. In both cases, though in different ways, we seek to draw conclusions about material processes from a purely syntactic formalism. In this final section, I will very briefly consider one well-known example of this: the "self-reproducing automata" of von Neumann (Burks *1966*; Arbib, this volume).

The basis of von Neumann's argument was the inference of the existence of a "universal constructor" from Turing's argument for a universal simulator (computer). On the purely formal side, von Neumann constructed a universe ("cellular space") of intercommunicating Turing machines arranged along some regular geometric array, like the cells in a multicellular organism, or the neurons in a brain, or atoms in a crystal. Each machine communicates with its nearest neighbors in the array. In the obvious fashion, the array as a whole changes "state" in time. The question was whether some sub-array ("tessellation automaton") could induce certain interesting behaviors in its complement, which could be *interpreted* as construction, replication, growth, development, evolution, and so on.

At the same time, von Neumann clearly believed that a "universal constructor" could exist as *hardware*. He envisaged equipping a Turing machine with sensors, so that it could "read" a blueprint, and with effectors, so that it could extract physical components from its environment, and assemble them as instructed by the blueprint being read. (An "effector", in this context, is a transducer from numbers to things, i.e., an "inverse meter".) The idea was that *both* computation and construction were algorithmic processes, and therefore whatever was true of the one must be true of the other.

Von Neumann also felt that the cellular spaces were not merely formal constructs, but actually comprised *models* of real-world constructors and their activities. In this, he was tacitly assuming Church's Thesis in its strongest form.

We have argued elsewhere (cf. Rosen *1985*), on grounds of causality, that any inference regarding a *material* universal constructor from the existence of a *formal* universal computer is unjustified. In that argument, we essentially showed that there was no intrinsic way of distinguishing between those input strings which encode "real-world information" and those which do not. Thus there was no way to distinguish between a computation which could be claimed to simulate a "real-world" process, and one which has no such realization.

Conversely, we can also see that the falsity of Church's Thesis means that there are aspects of material processes which cannot be formalized with any given encoding. Thus there are (a) formal constructions without material counterpart, and conversely, (b) material constructions without formal counterpart. Therefore, on both counts, von Neumann's argument is without material content, and merely involves the familiar equivocation on the terms "automaton" (or "machine") and "construction".

These considerations show how dangerous it can be to extrapolate unrestrictedly from formal systems to material ones. The danger arises precisely from the fact that computation involves only *simulation,* which allows the establishment of no congruence between causal processes in material systems and inferential processes in the simulator. We therefore lack precisely those essential features of encoding and decoding which are required for such extrapolations. Thus, although formal simulators can be of great practical and heuristic value, their theoretical significance is very sharply circumscribed, and they must be used with the greatest caution.

There are, of course, many other ramifications of Church's Thesis which we cannot touch on in this brief space. Its main role, as we have seen, is to separate out what is syntactic in a formal system from what is not; when the formal system is also a model of a material system, Church's Thesis does the same for causal relations. The Thesis in fact raises a host of deep questions about Natural Law, about causality, about modeling, and about the material realization of formalisms. Its central feature, the Turing machines, embody the essence of syntactics or string processing in a single, conceptually rich package. Even if (as I believe) Church's Thesis fails, it does so in a most instructive way. Its implications for the material sciences, and especially for biology, have barely begun to be explored.

References

Arbib, M.A.

1988 From universal Turing machines to self-reproduction. This volume.

Burks, A.

1966 *The Theory of Self-Reproducing Automata.* Urbana, IL: University of Illinois Press (1966).

Davis, M.

1958 *Computability and Unsolvability.* New York: McGraw-Hill (1958).

Gödel, K.

1931 Über formal unentscheidbare Sätze der Principia Mathematica und verwandte Systeme. *Monats. Math. Phys.* **38** (1931) 173–198.

Kleene, S.C.

1952 *Introduction to Metamathematics.* New York: von Nostrand (1952).

Myhill, J.

1966 *Creative Computation Revisited.* USAF Technical Report (1966).

Rosen, R.

1962 Church's Thesis and its relation to the concept of realizability in biology and physics. . *B. Math. Biol.* **24** (1962) 375–394.

1978 *Principles of Measurement.* New York: Elsevier (1978).

1985 *Anticipatory Systems.* New York: Pergamon Press (1985).

in press Biology and physics: An essay in natural philosophy (in press).

Turing, A.M.

1936-7 On computable numbers, with an application to the Entscheidungsproblem.*P. Lond. Math. Soc. (2)* **42** (1936-7) 230–265; A correction, ibid. **43** (1937) 544–546.