MONOGENIC NORMAL SYSTEMS ARE UNIVERSAL[1]

MICHAEL A. ARBIB

(received 24 August 1962)

1. Abstract

In 1943, Post conjectured that "monogenic normal systems are universal", and in 1961 Minsky proved a stronger result " 'tag' systems are universal" which implied the proof of Post's conjecture. The author had independently obtained a simple *direct* proof of Post's conjecture. The purpose of this note, then, is to present an exposition of Post's conjecture, and to show the full simplicity of its *direct* verification.

2.

The Hilbert school of Formalists placed great emphasis on finitistic axiomatic systems. Many successful attempts were made to characterise these systems in such a way that they themselves were amenable to mathematical (or, more properly, meta-mathematical) analysis. The various characterisations proved to be equivalent, and therein lay their power. In particular, the definitions of Turing and Post proved equivalent.

The approaches of Turing and Post have a great amount in common. First, we have an alphabet $a_0, a_1, a_2, \cdots$ in terms of which we can write the statements of our logic. A finite sequence of these symbols will be called a string. The result of juxtaposing the pair of strings X, Y will be written XY. Our logic L must have a designated finite set of strings — the *axioms* of L — together with a finite set of *rules of inference*. When $R(Y, X_1, \cdots, X_n)$ is a rule of inference of L, we say that Y is a consequence of $X_1, \cdots, X_n$, by R, in L.

A finite sequence of strings $X_1, X_2, \cdots, X_n$ is called a *proof* (of X_n) in a logic L if, for each i, $1 \leqq i \leqq n$, either

(1) X_i is an axiom, or

(2) there exist $j, \cdots, k < i$ such that X_i is a consequence of $X_j, \cdots, X_k$ in L by one of the rules of inference of L.

[1] This paper was presented to the Annual General Meeting of the Australian Mathematical Society in Sydney on August 17, 1962. It was supported in part by the U. S. Army Signal Corps, the Air Force Office of Scientific Research, and the Office of Naval Research; in part by the National Institute of Health (Grant B.-1865(C3)), and in part by the U. S. Air Force (Aeronautical Systems Division) under contract AF33(616)-7783.

We say that W is a *theorem* of L, or that W is *provable* in L, if there is a proof of W in L.

The only thing now lacking in our definition of our logic L is a restriction which ensures that the rules of inference operate in a strictly finitistic manner. The difference between the Turing and Post formalisms is in the way in which they effect this restriction.

Turing (1936) introduced the concept which has since been named, in his honour, Turing machine. A Turing machine comprises (cf. Fig. 1A):

(a) An infinite tape which is divided lengthwise into distinct squares; each of which may be blank, or bear a symbol from some finite alphabet. At any moment, all but a finite number of the squares must be blank.

(b) A "black box" which possesses a finite number of internal states, and which is capable — at any moment of time — of scanning one square of the tape, printing or erasing a symbol on that square, moving the tape one square to the left or right relative to the scanning device, and changing its internal state.

The operation of a Turing machine is deterministic, and proceeds on a quantised time scale, $t = 1, 2, 3, \cdots$. At any moment of time, the behaviour of the machine is uniquely determined by the internal state of the black box, and by the symbol scanned. (Note, however, that this determination will, in general, be different for different Turing machines).

Hence we may regard a Turing machine as a *set of quintuples* $q_i S_j S_k D q_l$, each such to be interpreted as: if the machine is in state q_i and the symbol scanned is S_j, then print S_k (where S_k may be a blank — corresponding to erasure), move tape D (where D may be: L, left one square; R, right one square; or N, not at all) and change the state of the machine to q_l. We make the convention that if the machine is in state q_i and the symbol scanned is S_j, and no quintuple $q_i S_j S_k D q_l$ is listed, then the machine is to cease operation.

The Turing form of the finitistic restriction, then, is to demand that a rule of inference R only be accepted for one of our logics L if there exists a Turing machine $T(R)$ such that Y is a consequence of $X_1, \cdots, X_n$ by R if and only if Y is printed on the tape by $T(R)$ when it eventually ceases operation after being started in some agreed initial state, say q_0, scanning the left-most symbol of the expression $X_1, X_2, \cdots, X_n$* printed on its tape. Let us agree to call one of our logics a *Turing system* if its rules of inference can be mediated by Turing machines, as above.

Post (1943) gave his finitistic restriction in terms of *productions*, each of the following form

* We can assume that the letters of the alphabet of L, and the comma will all belong to the set of symbols of $T(R)$.

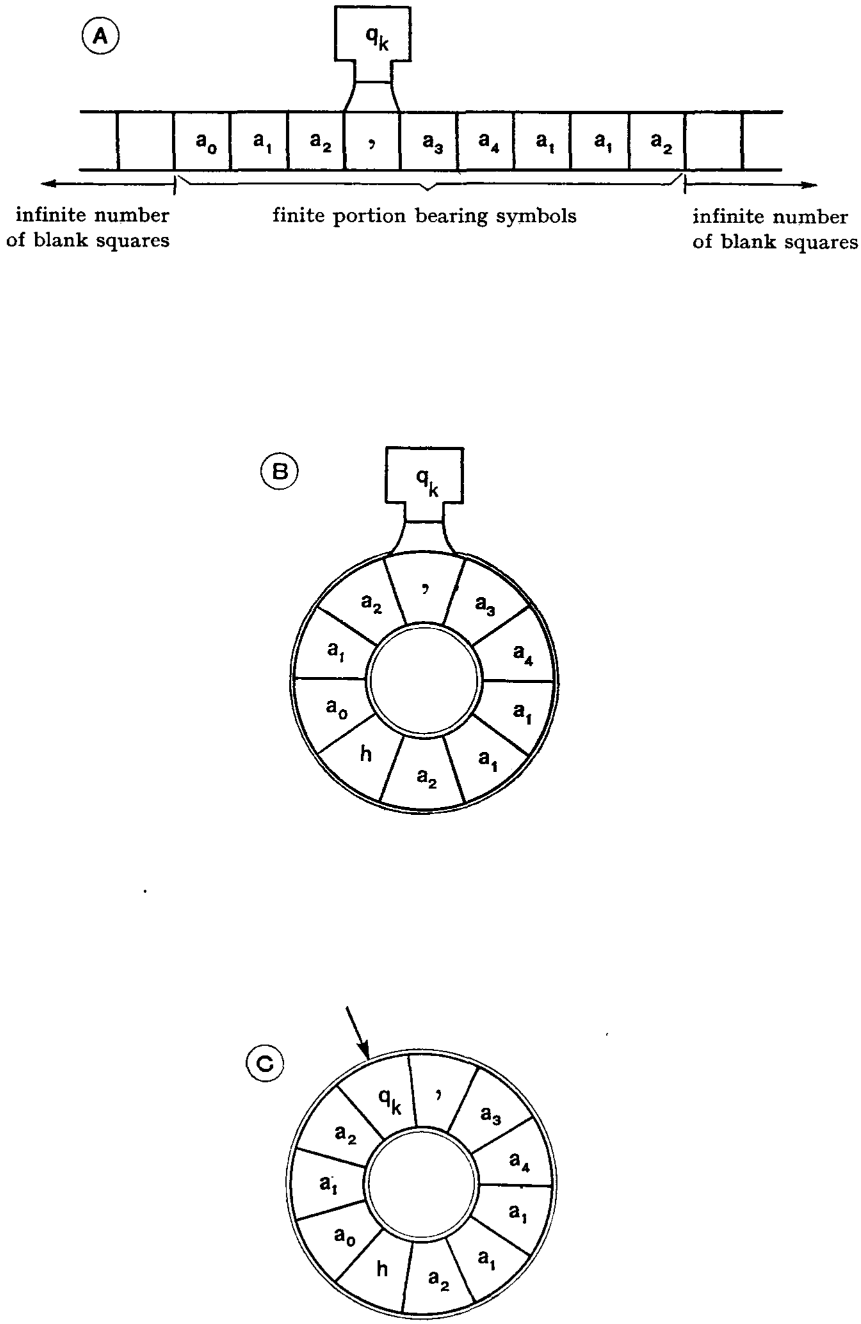

Figure 1

3 Portraits of a Turing Machine

https://doi.org/10.1017/S1446788700028330 Published online by Cambridge University Press

$$g_{11}P_1^{(1)}g_{12}P_2^{(1)}\cdots g_{1m_1}P_{m_1}^{(1)}g_{1(m_1+1)}$$
$$\cdots$$
$$g_{k1}P_1^{(k)}g_{k2}P_2^{(k)}\cdots g_{km_k}P_{m_k}^{(k)}g_{k(m_k+1)}$$
produce
$$g_1P_1g_2P_2\cdots g_mP_mg_{m+1}.$$

In this display, the g's represent specific strings including the null string, while the P's represent the operational variables of the production, and, in the application of the production, may be identified with arbitrary strings. We then add the restriction that each operational variable in the conclusion of the production is present in at least one of the premisses of the production, it having been understood that each premiss, and the conclusion, has at least one operational variable. Post called a logic in which each rule of inference is mediated by a production, a *system in canonical form.*

As we said above, the class of all Turing systems is equivalent to the class of all systems in canonical form (see, e.g. Davis (1958)).

Post (1943) also introduced 3 more restricted forms of logical systems:

(i) A system in canonical form is said to be in *normal form* if each of its productions is in the form

$$gP$$
produces
$$Ph$$

(ii) A normal system is *monogenic* if the g's of the premisses form a set $g_1, g_2, \cdots, g_k$ such that each string can be written in the form g_iP for at most one i.

(iii) A *"tag" system* is a monogenic normal system in which the g's constitute all sequences of some fixed length l, while the corresponding h's are identical for all g's having the same initial symbol.

Post proved that the normal systems are universal — i.e. any canonical system can be characterised by a normal system. *He then conjectured* (1943, p. 204) *that the monogenic normal systems were universal.* Minsky (1961) has recently given an elegant proof of the even stronger result: the "tag" systems are universal. Hence the weaker result of our title follows immediately. The purpose of this note is to give a *direct* proof of the weaker result, laying bare its full simplicity.

Minsky used the equivalence of the Post and Turing definitions to prove his result by showing how to represent an arbitrary Turing machine T as a "tag" system. Our task, then, is to provide the much simpler representation of a Turing machine T as a monogenic normal system $M(T)$.

Our construction results from a number of changes of viewpoint. First we may remove the infinitude of blank squares of T, agreeing to stick a new

https://doi.org/10.1017/S1446788700028330 Published online by Cambridge University Press

blank square on the tape whenever T tries to move off the end of its tape. Thus we may regard the tape as finite, and can replace the linear tape by a circular tape with a square marked h (assume this is a new symbol) to mark the ends of the linear tape it replaces (cf. fig. 1B). We can now signify our knowledge of which square is scanned and the state of T by inserting a new square on our circular tape immediately preceding the scanned square and writing on it the name of the internal state. We now replace T by an arrow! (See fig. 1C). *It is this use of a circular tape, and the replacement of the Turing machine by a state-square and an arrow, that constitutes the essence of our construction.* We can simulate the operation of T by rotating our circular tape until we encounter a q_j, and then operating according to the appropriate quintuple. In fact we shall apply the Turing machine state-symbol changes only when the arrow points to the square *preceding* that bearing the state-symbol.

To ensure monogenicity, we will have to rotate the tape more than one square at a time. In our construction, we shall cycle 2 squares at a time. However, if the arrow should point to a state-square, and the tape has an even number of squares, such 2-square cycling cannot bring the tape into 'the Turing position'. Hence we introduce a technical device which enables the machine to extricate itself from the above impasse: we insert an additional square after the square bearing the state symbol, and on it print the symbol *1* if the number of squares on the tape is even, and *0* if the number of squares on the tape is odd. This should be enough to motivate the following construction of a monogenic normal system.

If our Turing machine is to operate on the finite string of symbols P, then we get our monogenic normal system to operate on the string

$$hq_0\xi P \text{ where } \xi = \begin{cases} \mathit{0} \text{ if } P \text{ has an even no. of symbols} \\ \mathit{1} \text{ if } P \text{ has an odd no. of symbols.} \end{cases}$$

For cycling we introduce the 3 productions:

$$\alpha\beta P \rightarrow P\alpha\beta$$
$$q_j\mathit{1}\alpha P \rightarrow Pq_j\mathit{1}\alpha$$
$$q_j0P \rightarrow Pq_j0$$

(α, β indicate non-state symbols).

This corresponds to rotating the tape till the arrow points to the square preceding the one bearing a state symbol. This allows us to apply one of the following productions corresponding to the quintuples of the Turing machine. (S_0 denotes the blank).

$q_iS_jS_kLq_l$ gives rise to:

https://doi.org/10.1017/S1446788700028330 Published online by Cambridge University Press

$$S_r q_i \xi S_j \alpha P \to P q_l \xi S_r S_k \alpha$$

and

$$h q_i \xi S_j \alpha P \to P h q_l (1-\xi) S_0 S_k \alpha$$

$q_i S_j S_k N q_l$ gives rise to

$$\alpha q_i \xi S_j \beta P \to P \alpha q_l \xi S_k \beta P$$

$q_i S_j S_k R q_l$ gives rise to

$$\alpha q_i \xi S_j S_m P \to P \alpha S_k q_l \xi S_m$$
$$\alpha q_i \xi S_j h P \to P \alpha\, S_k q_l (1-\xi) S_0 h.$$

This system of productions is certainly a monogenic normal system $M(T)$. Further, if starting T in state q_0 scanning the left-most square of P causes T to stop in state q_k scanning the α of the expression $Q\beta\alpha R$ printed on its tape; then if we start $M(T)$ on the string $hq_0\xi P$ we will finally obtain $\beta q_k \eta \alpha R h Q$ ($\eta = 0$ or 1) from which we immediately obtain the desired result $Q\beta\alpha R$.

Thus we have indeed obtained a representation of our Turing machine T as a monogenic normal system $M(T)$. Hence, *monogenic normal systems are universal.*

Bibliography

[1] Davis, Martin, *Computability and Unsolvability*, McGraw-Hill, New York, 1958.

[2] Minsky, Marvin L., Recursive Unsolvability of the Problem of Tag, *Annals of Math.* **74**, (1961), 437–455.

[3] Post, Emil L., Formal Reductions of the General Combinatorial Decision Problem, *Amer. J. Maths.* **65** (1943), 197–215.

[4] Turing, A. M., On Computable Numbers with an Application to the Entscheidungsproblem, *Proc. London Math. Soc.* (2) **42** (1936) 230–265; with a correction, *Ibid.* **43** (1936-7), 544–546.

School of Mathematics,
University of New South Wales
and
Department of Mathematics and Research
Laboratory of Electronics,
Massachusetts Institute of Technology.

https://doi.org/10.1017/S1446788700028330 Published online by Cambridge University Press