5. Alan Turing and Computability

> Türing Machine. In 1936 Dr. Turing wrote a paper on the design and limitations of computing machines. For this reason they are sometimes known by his name. The umlaut is an unearned and undesirable addition, due, presumably, to an impression that anything as incomprehensible must be Teutonic. B. V. Bowden, *Faster than Thought*.

Mathematical foundations

On the morning of August 8, 1900 David Hilbert, professor of mathematics at the University of Göttingen and one of world's most eminent mathematicians, gave an invited lecture at the second International Congress of Mathematicians held in Paris during a summer which many chroniclers of this event said was hot and sultry. Instead of a conventional lecture giving a review of some area in which he had made a significant contribution, Hilbert gave a list of twenty-three problems to challenge mathematicians in the twentieth century. It was Hilbert's intention to place the whole of mathematics on a solid logical foundation. He ended his address with the following challenge: "This conviction of the solvability of every mathematical problem is a powerful incentive to the worker. We hear within us the perpetual call. There is the problem. Seek its solution. You can find it by pure reason, for in mathematics there is no *ignorabimus*."

Hilbert believed, as did his audience, that for any axiomatic system in mathematics the theorems derived should be complete, consistent and decidable. By *complete* he meant that a proof or a disproof could be found for every meaningful mathematical statement, by *consistent* that there could be no contradictions so that a statement could not be proven both true and false, and by *decidable* that for every meaningful mathematical statement there was a method for determining whether the statement was true or false. When Hilbert retired in 1930 at the age of sixty-eight, the city of Königsberg where he was born made him an honorary citizen at a meeting of the German Society of Scientists and Physicians which was being held there. Here he reiterated his convictions made thirty years before and ended his address with the words *Wir müssen wissen. Wir werden wissen* - We must know. We shall know. – words which were inscribed on his tombstone on his death in 1943.

We shall mention only three of Hilbert's twenty-three problems. The second was to prove the consistency of the axioms of the arithmetic of real numbers. The eighth was the Riemann hypothesis

concerning the real parts of the complex zeroes of the zeta function, a problem which has not yet been solved and which is considered to be the most famous unsolved problem in mathematics. The tenth problem was the decidability or decision problem although Hilbert appears to have restricted it to determining if Diophantine equations were solvable.

Just as Hilbert was being made an honorary citizen of Königsberg, Kurt Gödel, a young Czech-born mathematician working in Vienna, submitted a twenty-five-page paper to the *Monatshefte für Mathematik und Physik* which would put an end to Hilbert's dream of placing mathematics on a sound logical foundation. In it he showed that in any mathematical system, however simple, there would always be statements the truth or falsity of which could not be proven, i.e., that mathematics is incomplete. This in turn implied it could not be proven that mathematics is consistent. Only a few years later Hilbert's belief that every mathematical statement was decidable was shown to be unattainable. The solution to this problem which we shall discuss here is that proposed by the young Cambridge mathematics graduate Alan Turing who formulated a hypothetical mechanical means of solution and in so doing gave his name to the device now known as a Turing Machine. As Turing is now such a dominant figure in the history of computing, we shall give a very short summary of his brilliant yet short and tragic life and career.

Early life and education

Alan Mathison Turing was born on June 23, 1912 in a nursing home in Paddington, London, and was the second and last son of Julius and Sarah Turing. Julius Turing worked for the Indian Civil Service in the Madras Presidency; his wife, Ethel Sara Stoney, was the daughter of the chief engineer of the Madras railways. The Turings returned to England for the birth of their son. Julius soon returned to India but his mother remained in England until at least September 1913 when she rejoined her husband in India.

The Turings returned to England only rarely until Julius's retirement in 1926, and Alan and his brother John were cared for by guardians and were educated in private boarding schools. He found it difficult to fit into what was expected of him in the English public school system with the emphasis on a classical education and preparation for service to the British Empire and the relative neglect of mathematics and the sciences. His interest in science is said to have been stimulated by popular children's books on science, especially Edwin Tenney Brewster's *Natural Wonders Every Child Should Know* and later by Sir Arthur Eddington's *The Nature of the Physical World*. (A contemporary book for Canadian and American readers might have been *The Wonder Book of Knowledge* compiled and edited by Henry Chase Hill which presented "The marvels of modern industry and invention. The interesting stories of common things. The mysterious processes of nature simply explained.")

Turing entered King's College, Cambridge in 1931 where he used the liberal atmosphere to satisfy his wide-ranging intellectual interests which included reading John von Neumann's recently published book

on quantum mechanics. He also had the opportunity to express more openly his homosexuality of which he became increasingly aware in his school days. He graduated from King's in 1934, and was elected a Fellow in 1935 for a thesis on the Central Limit Theorem which he established independently and without knowledge of the proof published about twelve years earlier. It was in 1935 that he first heard of Hilbert's decision problem in a course given by the topologist M. H. A. (Max) Neuman.

In 1936 he enrolled as a graduate student at Princeton and received his Ph.D. in 1938 for a thesis on algebra and number theory. He was offered a temporary post at Princeton by von Neumann but declined it, and returned to Britain where he resumed his Fellowship at Cambridge and continued the work he had begun at Princeton. He also in secret worked part-time for the British cryptographic department, the Government Code and Cypher School as it was formally known. He was the first scientist to be engaged in such work as previous positions were held by arts graduates and chess players.

Bletchley Park

Immediately on the outbreak of the Second World War in September 1939 Turing took up a full-time position at the cryptographic headquarters at Bletchley Park in Buckinghamshire. His work on deciphering German messages encrypted by the Enigma machine resulted first in the generalization of the Polish Bombe, an electromechanical device for decrypting Luftwaffe traffic. He then developed methods for decrypting the much more difficult Enigma naval traffic. This latter work was said to have shortened the war by months or possibly years and saved countless Allied lives. For his work at Bletchley Park Turing was awarded the Order of the British Empire.

National Physical Laboratory and Manchester

At the end of the war Turing joined the staff of the National Physical Laboratory at Teddington to work on the design of an Automatic Computing Engine or ACE, a stored-program electronic computer to rival the proposed American EDVAC. He not only made detailed plans for the design of this machine but also for its programming including a suggestion for an Abbreviated Code Instructions scheme which could be considered a very early programming language. However none of Turing's work had any influence on the project due in part to bureaucratic indifference and probably to Turing's inability to work well in a governmental environment. In October 1947 he returned, with some encouragement from NPL, to Cambridge for the coming academic year.

In May 1948 Max Newman, his former Cambridge teacher who was now Director of the Royal Society Computing Laboratory at Manchester, invited Turing to fill the rather ill-defined position of Deputy Director. Turing worked on program development and explored various topics in mathematical logic and game playing. Some of his work on this latter topic is given in a short paper in the symposium

edited by Lord Bowden whose tongue-in-cheek comment on the Turing name is given at the beginning of this chapter. He also wrote a manual on machine-language programming for the Manchester computer, part of which served as a manual for FERUT, the model of the Ferranti computer which was installed at the University of Toronto in the 1950s. In July 1951 Turing was made a Fellow of the Royal Society.

In March 1952 Turing was arrested and charged with "gross indecency" because of a homosexual affair with a young man. Rather than be imprisoned he agreed to a one-year period of oestrogen injections to neutralize his libido. On June 8, 1954 he decided that the world was too much for him, and he committed suicide by biting into an apple laced with cyanide.

Turing Machines

As has been mentioned already Turing first heard of the decision problem in Max Neuman's lectures in which it was suggested that a solution might be found by a mechanical method, on paper at least, which could decide if any mathematical assertion is provable. Turing was struck by the term "mechanical" which implied to him a procedure that could be carried out by either a machine or a person following a prescribed list of rules. He began by defining "computable numbers" as those real numbers, i.e., positive and negative integers and rational and irrational numbers, that could be expressed as infinite decimals and computed by some given set of rules. Examples are $1/4 = 0.25000...$, $1/6 = 0.16666...$ and $\sqrt{2} = 1.41421...$. He then devised a hypothetical machine that would carry out these operations which he maintained encompassed all "the possible processes which can be carried out in computing a number". We shall describe such a machine, now known as a Turing Machine, and give a few simple examples before discussing the theoretical importance of this work..

The machine imagined by Turing was simply a device which could read sequentially, one at a time, the squares on an infinitely long paper tape marked off into squares, changing or leaving unchanged the symbol on the square and then moving one square left or right. The two important features were that there were only a finite number of symbols and a finite number of "states" which determined the action of the machine on reading a symbol. Thus the machine would start in some initial state reading the symbol on a given square, then change or erase the symbol, and then move one square left or right and possibly change to another state. This process of reading and writing, moving from square to square and changing from state to state would continue until the machine would pass into a stopping state and stop. Thus the operation of a Turing Machine could be specified by a number of quintuples of the form

(Present state, symbol read, symbol written, move left or right, new state) .

We shall represent the states by S_0, S_1, S_2, S_3, ... where S_0 is the stopping state, and the moves left and right by `L` and `R` respectively. (There is no loss of generality in limiting the possible symbols to the digits `0` and `1`, although some definitions of Turing Machines allow only one symbol while others allow

several symbols such as the decimal digits 0, 1, 2, ..., 9.) For example, the quintuple (S_2,1,0,R,S_4) would mean that if the machine is in state S_1 and is reading the symbol 1, it would write the symbol 0, move the read-write head one square to the right, and go to state S_4. The quintuple (S_2,1, ,R,S_4) would have the same meaning except that the symbol written would be a blank, i.e., the symbol 1 on the tape would be erased. We shall assume that the machine is initially in state S_1 and is reading the leftmost marked square on the tape.

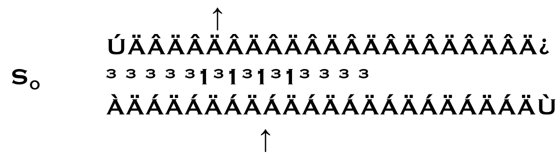
As a very simple example consider a Turing Machine which reads a sequence of consecutive 1s on a tape until it reads the blank square immediately to the right of the last 1. It then writes a 1 and starts moving to the left and finally stops on the leftmost marked square. This Turing Machine is defined by the following four quintuples:

(S_1,1,1,R,S_1)   (S_1, ,1,L,S_2)

(S_2,1,1,L,S_2)   (S_2, , ,R,S_0)

The following diagram shows the operation of the machine starting with a tape marked with the initial configuration 111. At each stage the current state of the machine is shown to the left of the tape while the position of the read-write head is indicated by the arrow ↑ under the appropriate square.

```
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S1      3 3 3 3 31313131 3 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                 ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S1      3 3 3 3 31313131 3 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                   ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S1      3 3 3 3 31313131 3 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                     ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S1      3 3 3 3 31313131 3 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                       ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S2      3 3 3 3 3131313131 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                     ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S2      3 3 3 3 3131313131 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                   ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S2      3 3 3 3 3131313131 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
                 ↑
        ÚÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄÂÄ¿
S2      3 3 3 3 3131313131 3 3 3
        ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
```



If the non-negative integer N is represented by N+1 consecutive 1s, then the above Turing Machine program represents the addition of 1 using 2 +1 = 3 as test data.

As a second example let us consider a Turing Machine that reads a sequence of 0s and 1s on the tape and replaces the sequence by the parity bit defined as 0 when the number of 1s is even and 1 when the number of 1s is odd. The machine may be defined by the following seven quintuples:

(S_1,0, ,R,S_1) (S_1,1, ,R,S_2) (S_1, ,0,R,S_3)

(S_2,0, ,R,S_2) (S_2,1, ,R,S_1) (S_2, ,1,R,S_3)

(S_3, , ,L,S_0)

A little thought will show that state S_1 indicates that an even number, including zero, of 1s has been read, and state S_2 that an odd number of 1s has been read. Thus as the read-write head scans successive squares erasing the symbols as it moves, its internal state alternates when necessary between S_1 and S_2. The third quintuple in each row writes the correct parity bit, and the seventh and final quintuple stops the machine.

ÀÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÁÄÙ
↑

As a final example, which will left as an exercise, the following quintuples define a Turing Machine which determines if a sequence of `0`s and `1`s is a palindrome, i.e., reads the same forwards and backwards. For example, `100101001` is a palindrome, and `1010001` is not. If the sequence is a palindrome, a single `1` is written on the tape, the original sequence having been erased. If the sequence is not a palindrome, `a 0` is written on the tape immediately to the right of any portion of original sequence remaining after execution of the program has stopped.

$(S_1,0,0,R,S_2)$	$(S_1,1,1,R,S_3)$	$(S_1,\ ,1,R,S_9)$
$(S_2,0,0,R,S_2)$	$(S_2,1,1,R,S_2)$	$(S_2,\ ,\ ,L,S_4)$
$(S_3,0,0,R,S_3)$	$(S_3,1,1,R,S_3)$	$(S_3,\ ,\ ,L,S_5)$
$(S_4,0,\ ,L,S_6)$	$(S_4,1,1,R,S_8)$	
$(S_5,0,0,R,S_8)$	$(S_5,1,\ ,L,S_6)$	
$(S_6,0,0,L,S_6)$	$(S_6,1,1,L,S_6)$	$(S_6,\ ,\ ,R,S_7)$
$(S_7,0,\ ,R,S_1)$	$(S_7,1,\ ,R,S_1)$	$(S_7,\ ,1,R,S_9)$
		$(S_8,\ ,0,R,S_9)$
		$(S_9,\ ,\ ,L,S_0)$

An interesting treatment of this example is given in Andrew Hodges' *New Scientist* article cited in the References.

Computability

Turing devised a scheme for the numerical encoding of the quintuples defining any machine so that it could be assigned a unique decimal description number which would be an integer, albeit a very large one. Therefore the description number for any Turing Machine could be written on a tape one digit per square and then followed by a number representing input to the machine thus encoded. Turing then developed what is now termed a Universal Turing Machine that would take such a tape and by interpreting the instructions for the particular machine written on the tape and using the data to the right of its description number perform the same calculations as would the given Turing Machine working on the given data.

Turing then imagined that the description numbers for all possible Turing Machines, including those corresponding to computable numbers, be sorted in numerical order. A subset of these numbers could be termed "satisfactory" description numbers corresponding to those machines for generating computable numbers. However Turing showed that there was no Turing Machine capable of deciding whether a given description number is satisfactory. Suppose on the contrary, he said, that there was such a machine. Then one could construct a new Turing Machine which finds the first digit of the first satisfactory

description number and prints a different first digit, then finds the second digit of the second satisfactory description number and prints a different second digit, etc. Thus it generates an infinite computable number that has a satisfactory description number whatever it may be. Yet the number is different from all of the numbers generated by every Turing Machine with a satisfactory description number. Thus the hypothesis that there does exist a Turing Machine capable of determining if the description number of a given Turing machine is satisfactory is false. This in turn answers Hilbert's decidability problem in the negative, i.e., it is not possible to determine if every mathematical statement is true or false. (There are other ways of proving the same result including one preferred by Turing in which a machine acted upon its own description number.)

Turing showed his work to Max Neuman in April 1936 just as the American logician Alonzo Church established the same results using another method known as the λ-calculus. Publication of Turing's work was delayed a few months as he was modifying his paper to acknowledge Church's work and showing that the two methods were equivalent. His classic paper "On computable numbers, with an application to the *Entscheidungsproblem*" was published later the same year in the *Proceedings of the London Mathematical Society*.

A play and a novel

In the 1980s the British playwright Hugh Whitemore wrote *Breaking the Code*, a play on Turing's life and work. It was based, as stated on the cover, on Andrew Hodges's definitive biography *Alan Turing: The Enigma* published earlier in the decade. (The play was subsequently made into a televison drama.) The play discusses Turing's work on the *Entscheidungsproblem* at Cambridge, his work at Bletchley Park on codebreaking, and his computer work at Manchester. It also gives a sympathetic treatment of his homosexual encounters and the resulting problems with officialdom. In one scene Turing, in a discussion with the classicist and wartime cryptographer Dilwyn Knox, discusses Hilbert's work and defines consistency, completeness and decidability ("Consistency means that you won't ever get a contradiction in your system; in other words, you'll never be able to follow the rules of your system and end up showing that two and two make five."). He then mentions Gödel's theorem that a mathematical system is either inconsistent or incomplete. ("It's a beautiful theorem, quite beautiful, I think Gödel's theorem is the most beautiful system I know.") Later in the same scene Knox's says to Turing: "Forgive me for asking a crass and naïve question – but what is the point of devising a machine that cannot be built in order to prove that there are certain mathematical statements that cannot be proved?" In another scene Knox ends a discussion with the following quotation by the logician Ludwig Wittgenstein which raises issues far beyond any related to Turing's great accomplishments: "We feel that even when all scientific questions have been answered, the problems of life remain completely unanswered."

Robert Harris, a Cambridge graduate and former television and newspaper journalist, is the author of several works of nonfiction but is probably better known to the reading public as the author of about half a dozen thrillers some of which are set in Roman times. The theme of his 1995 novel *Enigma*, as the title may suggest, is concerned with the breaking of the Enigma code and is centred at Bletchley Park. The central character is Tom Jericho, a student of Alan Turing's, who was sent to Bletchley Park at the outbreak of the war. The novel opens in 1943 as Jericho returns to his Cambridge college to recover from a nervous breakdown, He is soon recalled to Bletchley Park to help break the Enigma code. There is a secondary plot involving Jericho's former girlfriend who is missing and is suspected of being a German agent, especially by Jericho who finds stolen cryptograms hidden in her room. By the end of the book the Enigma code has been broken and the secondary plot resolved in a quite unexpected but plausible manner. Harris shows a good understanding of the administrative and physical layout of Bletchley Park and gives sufficient technical detail of codebreaking without letting it intrude on the plot. He also gives an excellent portrayal of life in wartime Britain with the almost unbearable coldness of the living and working conditions, the lack of food, and the awfulness of what food that was available.

Both the *Breaking the Code* and *Enigma* could serve as excellent introductions to Turing's work and might encourage some readers to turn to some of non-fictional accounts of wartime codebreaking, five of which are referred to at the beginning of Harris's book.

A Turing Machine simulator

The sample programs given earlier in the chapter have been executed on a Turing Machine simulator written in **J** and given at the end of this section. The following dialogue shows its use to run the first example of the addition of `1` to an integer. The global variable `TRACE` gives the state number, position of the read-write head and tape configuration for each quintuple executed.

```
   tm01=: 'S1';'1';'1';'R';'S1'
   tm02=: 'S1';' ';'1';'L';'S2'
   tm03=: 'S2';'1';'1';'L';'S2'
   tm04=: 'S2';' ';' ';'R';'S0'
   tm0=: 4 5 $ tm01, tm02, tm03, tm04
   tm0 TM '1111'
   tm0 TM '111'
ÚÄÂÂÂÂÄÄÂÂÂÄÄÂÂÂÄÄÂÂÂÂÄÂ¿
³ ³ ³ ³ ³1³1³1³1³ ³ ³ ³ ³
ÀÄÁÁÁÄÁÁÁÄÁÁÁÄÁÁÁÄÁÁÁÁÄÙ
  TRACE
S1  4     111
S1  5     111
S1  6     111
S1  7     111
S2  6     1111
S2  5     1111
```

```
S2 4    1111
S2 3    1111
S0 4    1111
```

```
NB.    Turing Machine
NB.    Keith Smillie
NB.    Department of Computing Science
NB.    University of Alberta
NB.    Edmonton, Alberta T6G 2H1
NB.    August 2011

NB. Simulation of a two-symbol Turing Machine with the symbols
NB. '0' and '1' and ' ' for blanks. The states are represented
NB. by 'S0', 'S1', 'S2', ...  with 'S0' being the stopping state.
NB. The machine starts in state 'S1' reading the leftmost marked
NB. square and continues reading and writing one square at a time
NB. and moving one square left or right ('L' or 'R') according
NB. to the given quintuples until computation stops.
NB.
NB. The verb TM is dyadic with a table of quintuples defining the
NB. machine as the left argument and the initial tape with no
NB. leading or trailing blanks as the right argument. (Four blanks
NB. are appended to each end of the tape at the beginning of
NB. execution of the program. The result is the configuration of
NB. the tape at the end of the computation. The global variable
NB. TRACE gives the sequence of tape configurations as a table
NB. with each row preceded by the corresponding state and the
NB. pointer position.

NB. Ex0. Add 1 (tm0 TM '1111')
tm01=: 'S1';'1';'1';'R';'S1'
tm02=: 'S1';' ';'1';'L';'S2'
tm03=: 'S2';'1';'1';'L';'S2'
tm04=: 'S2';' ';' ';'R';'S0'
tm0=: 4 5 $ tm01, tm02, tm03, tm04

NB. Ex1. Parity digit (tm1 TM '10011101'))
tm11=: 'S1';'0';' ';'R';'S1'
tm12=: 'S1';'1';' ';'R';'S2'
tm13=: 'S1';' ';'0';'R';'S3'
tm14=: 'S2';'0';' ';'R';'S2'
tm15=: 'S2';'1';' ';'R';'S1'
tm16=: 'S2';' ';'1';'R';'S3'
tm17=: 'S3';' ';' ';'L';'S0'
tm1=: 7 5 $ tm11,tm12, tm13, tm14, tm15, tm16, tm17

NB. Ex2 Sum (tm2 TM '111 1111')
tm21=: 'S1';'1';'1';'R';'S1'
tm22=: 'S1';' ';'1';'L';'S2'
tm23=: 'S2';'1';'1';'L';'S2'
tm24=: 'S2';' ';' ';'R';'S3'
```

```
tm25=: 'S3';'1';' ';'R';'S4'
tm26=: 'S4';'1';' ';'R';'S0'
tm2=: 6 5 $ tm21, tm22, tm23, tm24, tm25, tm26

NB. Palindrome (tm3 TM '100101001')
tm301=: 'S1';' ';'1';'R';'S9'
tm302=: 'S1';'0';'0';'R';'S2'
tm303=: 'S1';'1';'1';'R';'S3'
tm304=: 'S2';' ';' ';'L';'S4'
tm305=: 'S2';'0';'0';'R';'S2'
tm306=: 'S2';'1';'1';'R';'S2'
tm307=: 'S3';' ';' ';'L';'S5'
tm308=: 'S3';'0';'0';'R';'S3'
tm309=: 'S3';'1';'1';'R';'S3'
tm310=: 'S4';'0';' ';'L';'S6'
tm311=: 'S4';'1';'1';'R';'S8'
tm312=: 'S5';'0';'0';'R';'S8'
tm313=: 'S5';'1';' ';'L';'S6'
tm314=: 'S6';' ';' ';'R';'S7'
tm315=: 'S6';'0';'0';'L';'S6'
tm316=: 'S6';'1';'1';'L';'S6'
tm317=: 'S7';' ';'1';'R';'S9'
tm318=: 'S7';'0';' ';'R';'S1'
tm319=: 'S7';'1';' ';'R';'S1'
tm320=: 'S8';' ';'0';'R';'S9'
tm321=: 'S9';' ';' ';'L';'S0'
tm3a=: tm301,tm302,tm303,tm304,tm305,tm306,tm307
tm3b=: tm308,tm309,tm310,tm311,tm312,tm313,tm314
tm3c=: tm315,tm316,tm317,tm318,tm319,tm320,tm321
tm3=: 21 5 $ tm3a,tm3b,tm3c

TM=: 3 : 0
:
Quintuples=: x.
Tape=: '    ', y.,'    '
Ptr=. 4
State=. 'S1'
TRACE=: (1,6+#Tape) $ (2{.State), (3.0":Ptr),'   ' ,Tape
whilst. -. (State -: 'S0') do.
   Symbol=. Ptr{Tape
   c0=. >0{"1 Quintuples
   c1=. >1{"1 Quintuples
   Q=. ,((*./"1 State e."1 c0) *. (Symbol = c1)) # Quintuples
   'Symbol Move State'=. 2 3 4{Q
   Tape=: Symbol Ptr} Tape
   Ptr=. Ptr + (Move = 'R') { _1 1
   TRACE=: TRACE, (2{.State), (3.0":Ptr),(2$' '),Tape
end.
<"0 Tape
)
```