4.5 A model of computation
Mark Scheme

Mark schemes

Q1.

Mark is for AO1 (understanding)

The Halting problem is non-computable / undecideable // there is no algorithm that solves the Halting problem;
A. it is not computable

In general, inspection alone cannot always determine whether any given algorithm will halt for its given inputs // a program cannot be written that can determine whether any given algorithm will halt for its given inputs;

Max 1 mark

[1]

Q2.

All marks AO1 (knowledge)

Finite set of states (in a state transition diagram);
A set of transition rules;
A (sensing) read-write head (that can move along the tape one square at a time);
Start state;
(Set of) accepting / halting states;
State register // current state;

[2]

Q3.

All marks AO1 (knowledge)

A Turing machine that can execute/simulate the behaviour of any other Turing machine // can compute any computable sequence;

Faithfully executes operations on the data precisely as the simulated TM does;
(Note: must have idea of same process)

Description of/Instructions for TM (and the TM's input) are stored on the (Universal Turing machine's) tape // The UTM acts as an interpreter; **A**. take any other TM and data as input

Alternative definition:
A UTM, U, is an interpreter that reads the description <M> of any arbitrary Turing machine M;

and faithfully executes operations on data D precisely as M does.;

The description <M> is written at the beginning of the tape, followed by D.;

Max 2 marks

[2]

Q4.

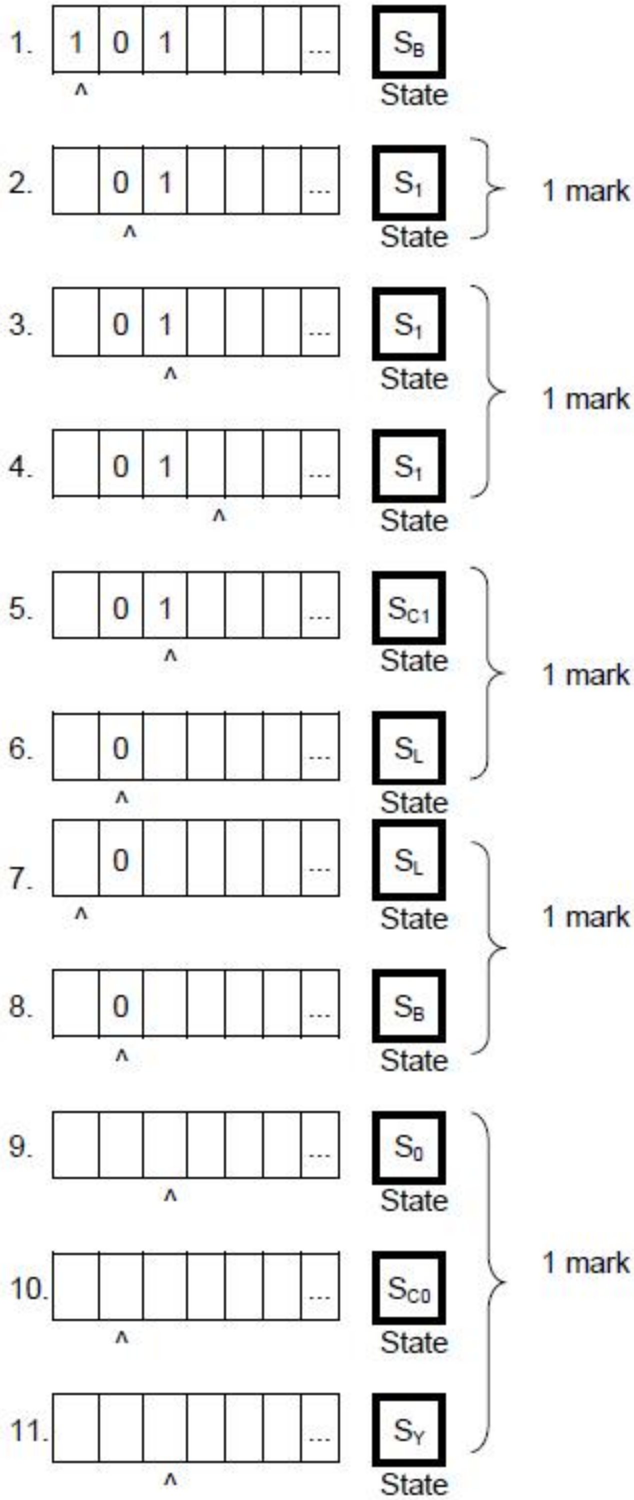
Mark is for AO1 (understanding)

Because it has an infinite amount of memory / tape;

[1]

Q5.

(a) One mark per bracketed section.



Must have correct tape contents and state for each mark
A. Blank symbols instead of empty cells
DPT If the read/write head is not drawn on some rows, this should result in the

loss of the mark on the first occasion that it is missing only. Marks should be awarded for subsequent rows, even if the read/write is not drawn.

5

(b) (After a 0 has been read,) the rules keep moving the read/write head to the right (preserving the contents of the tape) // the read/write head skips right over 0s and 1s;
Until a blank symbol is encountered / the end of the number is reached, then the state is changed to S_{C0} (and the head is moved left/direction reversed);
Note: To achieve the first mark, it must be clear that the head moves right regardless of whether a 0 or 1 is read and also that this is a repeated process ie not just moving one place right.
Note: If it is stated that the process of moving continues until the end of the number is reached, then it can be inferred that the head was moving right for the first mark, if this was not explicitly stated.
Note: Marks should not be awarded for just explaining what the rules do individually.

2

[7]

Q6.

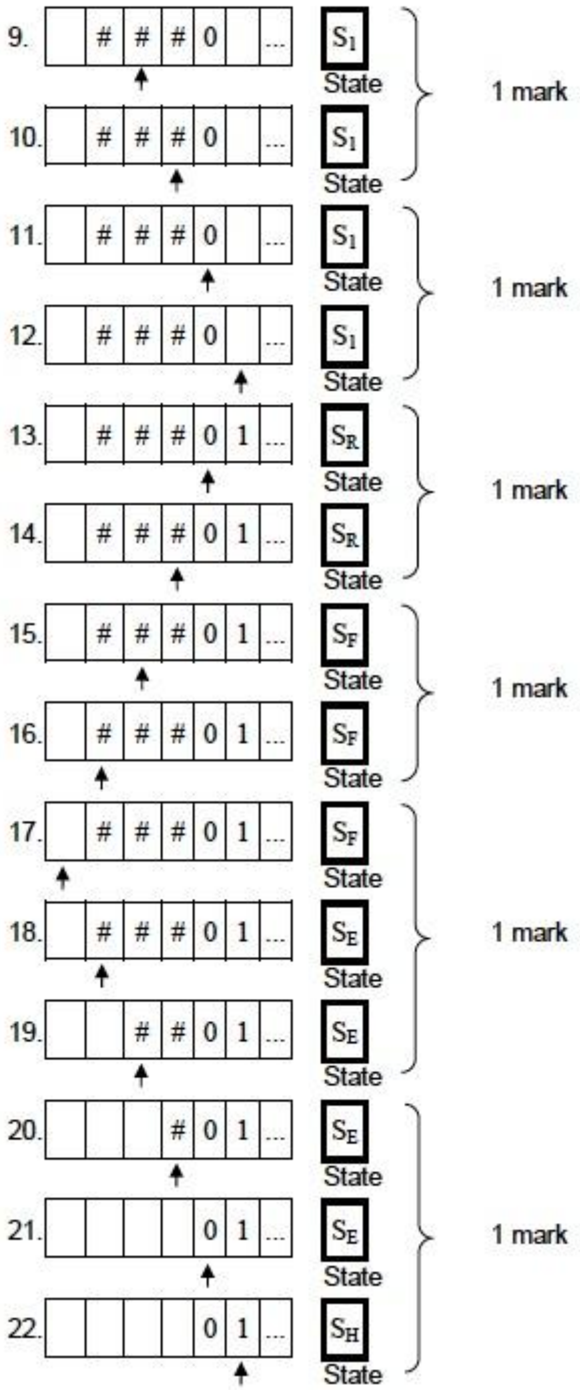
(a)

Turing machine component	**Number (1-5) of modern computer system component with most similar role**
Transition function	5; **A.** Program
Tape	3; **A.** Main Memory / Memory

2

(b) If (and only if) an algorithm exists to solve a problem then a Turing machine can be designed to solve the problem;
A. Statement made in reverse i.e. "if a Turing machine exists..."
A. Statement made as a negative i.e. "if no algorithm exists ..."
A. A Turing machine can compute any algorithm // any algorithm can be computed by a Turing machine

1



(c)

Must have correct tape contents and state for each mark
A. Blank symbols instead of empty cells
A. Content written anywhere on the tape as long as the position is correct relative to the read/write head
A. Read/write head drawn off left hand end of tape at stage 17 if tape contents are written at left hand end of tape

DPT If the read/write head is not drawn on some rows, this should result in the loss of the mark on the first occasion that it is missing only. Marks should be awarded for subsequent rows, even if the read/write head is not drawn.

6

(d) To reverse a (binary) string/number // to produce a copy of a

(binary) string/number with the order of the characters/digits reversed;
R. Flips bits, but **A.** Flips order of bits
A. Mirror the input

1

[10]

Q7.

(a)

Current State	S_3	S_3	S_3
Input Symbol	a	b	c
Next State	S_6	S_6	S_4

1 mark for all six correct values in the bold rectangular area
The columns do not have to be in the same order as shown, but the pairings must be correct i.e. (a - S_6, b - S_6, c - S_4).
A 4 for S_4 and 6 for S_6

1

(b) S_3
A 3
I An additional name given to the state eg "State 3"

1

(c) To ensure that a non-valid string is trapped // prevent the accepting state being reached;
A To capture invalid input
A To capture strings that are too long / have extra characters
NE Infinite loop / state cannot be left

1

(d) `a(bc)*a` // `a(bc)+?a` // `(aa)|(a(bc)*a)` // `(aa)|(a(bc)+?a)` // `(aa)|(a(bc)+a)`

1 mark for recognising an a at both ends
1 mark for correctly recognising 0 or more repetitions of bc
I `^` and `$` at start and end of expression
A Any type of bracket

2

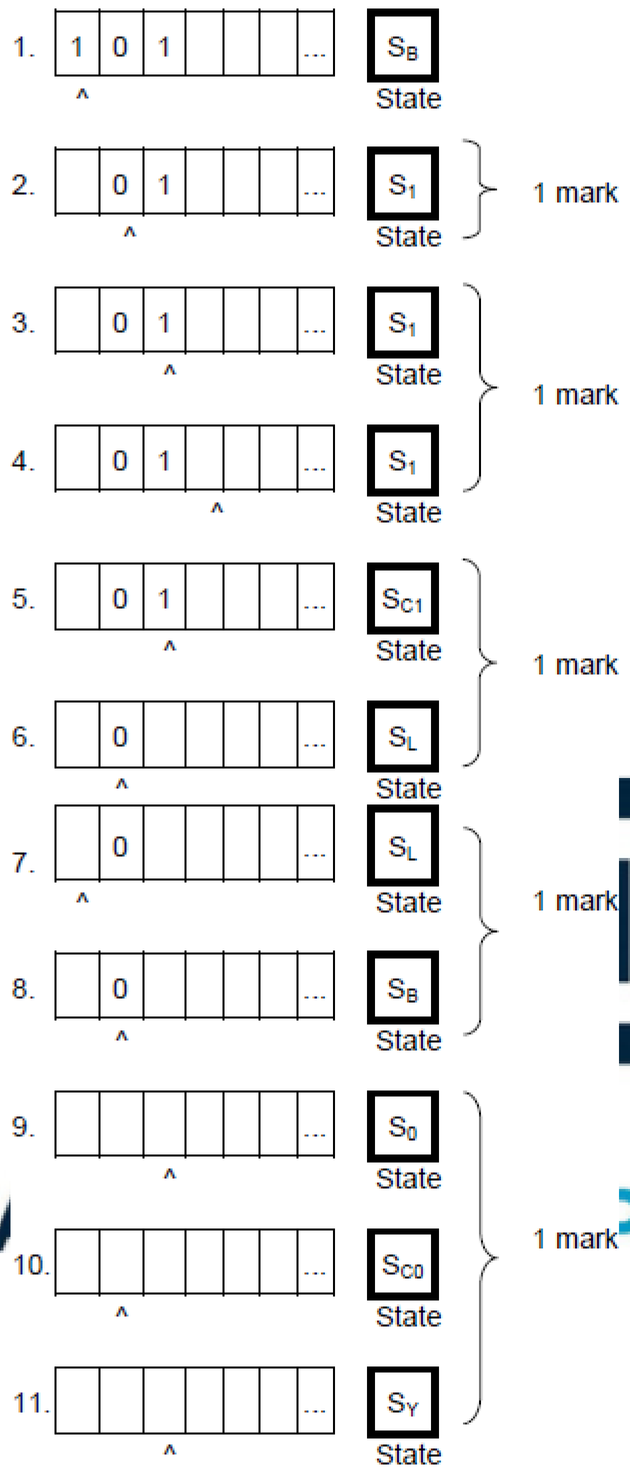
(e) Turing Machine has (an infinite / unlimited amount of) memory / storage;
Turing Machine can read <u>and</u> write / input <u>and</u> output (data) to / from a tape;
Turing Machine has <u>infinitely long</u> tape;
NE Turing Machine has a tape
MAX 1

1

[6]

Q8.

(a) **1 mark** per bracketed section.



Must have correct tape contents and state for each mark
A blank symbols instead of empty cells
DPT If the read / write head is not drawn on some rows, this should result in the loss of the mark on the first occasion that it is missing only. Marks should be awarded for subsequent rows, even if the read / write head is not drawn.

5

(b) (After a 0 has been read,) the rules keeps moving the read / write head to the right (preserving the contents of the tape);
Until a blank symbol is encountered / the end of the number is reached, then the state is changed to S_{C0} (and the head is moved left / direction reversed);
Note: To achieve the first mark, it must be clear that the head moves right regardless of whether a 0 or 1 is read and also that this is a repeated process

ie not just moving one place right.
Note: If it is stated that the process of moving continues until the end of the number is reached, then it can be inferred that the head was moving right for the first mark, if this was not explicitly stated.
Note: Marks should not be awarded for just explaining what the rules do individually.

2

(c) It reads instructions one at a time / / reads instructions in sequence / / deals with instructions line by line;
And executes these instructions;
Instructions are / transition function is stored on the tape;
A "rules" for "instructions"
MAX 2

2

[9]

Q9.

(a) (i) ❷ S_1 **A** 1, State 1
❸ S_T **A** T, State T
Both answers correct to get mark;

1

(ii) δ (S_B, 0) = (S_0, x,⟶);

A 0, x, ⟶ or 0 | x | ⟶;
R if additional rules listed
I minor transcription errors e.g. missing , (δ

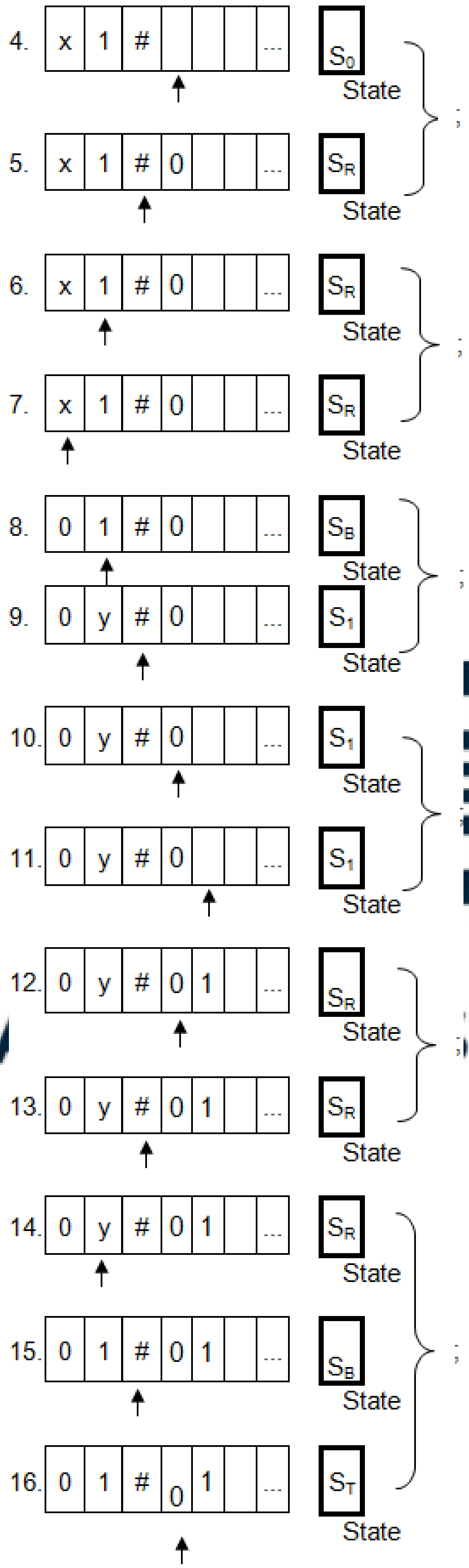
1

(iii) δ (S_R, x) = (S_B, 0,⟶) and δ (S_R, y) = (S_B, 1, ⟶);

A x, 0, ⟶ or x | 0 | ⟶ and y, 1, ⟶ or y | 1 | ⟶
R if additional rules listed
I minor transcription errors e.g. missing , (δ

1

(b) One mark per bracketed section.



Must have correct tape contents and state for each mark

A blank symbols instead of empty cells

DPT If the read / write head is not drawn on some rows, this should result in the loss of the mark on the first occasion that it is missing only.
Marks should be awarded for subsequent rows, even if the read / write head is not drawn.

6

(c) (i) Mark symbol currently being copied // to indicate how much of the string has been copied so far // to indicate where to return to (to copy next symbol);
A placeholders
NE x represents 0, y represents 1

1

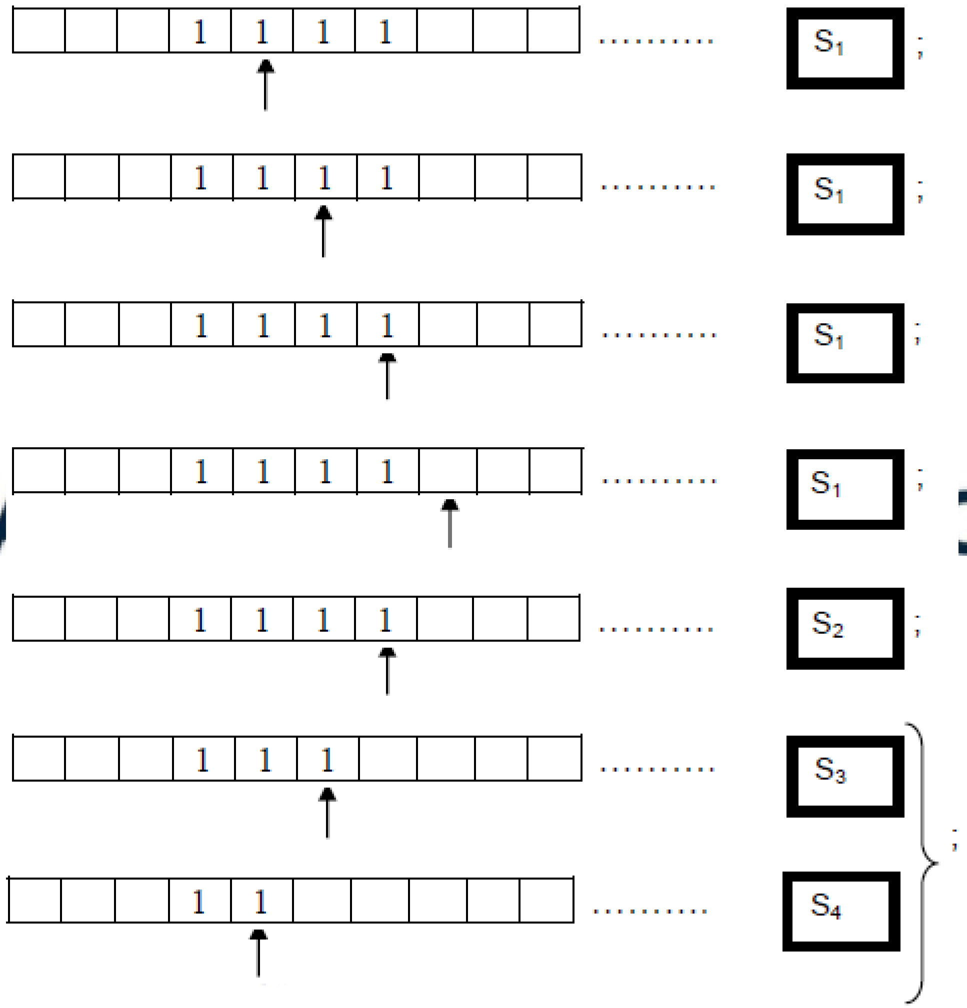
(ii) Copy a string // copy a binary number // copy a bit pattern;
A Repeat

1

[11]

Q10.

(a)



1 mark for each of the top five rows
1 mark for sixth and seventh row together
Must have correct tape contents and state for each mark

A the blank cell symbol □ in blank cells
A answers in which the initial situation of the TM is repeated

A If the read/write head is not drawn on some rows, this should result in the loss of the mark on the first occasion that it is missing only.

Marks should be awarded for subsequent rows, even if the read / write head is not drawn.

6

(b) Deletes two ones from the (right hand) end of the string // Subtracts two from a (unary) number;
A bits for ones
R end of tape for end of string
NE deletes two ones

1

(c) A Turing machine that can execute/simulate the behaviour of any other Turing machine // can compute any computable sequence;
Faithfully executes operations on the data precisely as the simulated TM does;
(Note: Must have idea of same process)
Description of/Instructions for TM (and the TM's input) are stored on the (Universal Turing machine's) tape // The UTM acts as an interpreter;
A take any other TM and data as input

Alternative definition:

A UTM, U, is an interpreter that reads the description <M> of any arbitrary Turing machine M; and faithfully executes operations on data D precisely as M does.; The description <M> is written at the beginning of the tape, followed by D.;

Max 2

[9]

Q11.

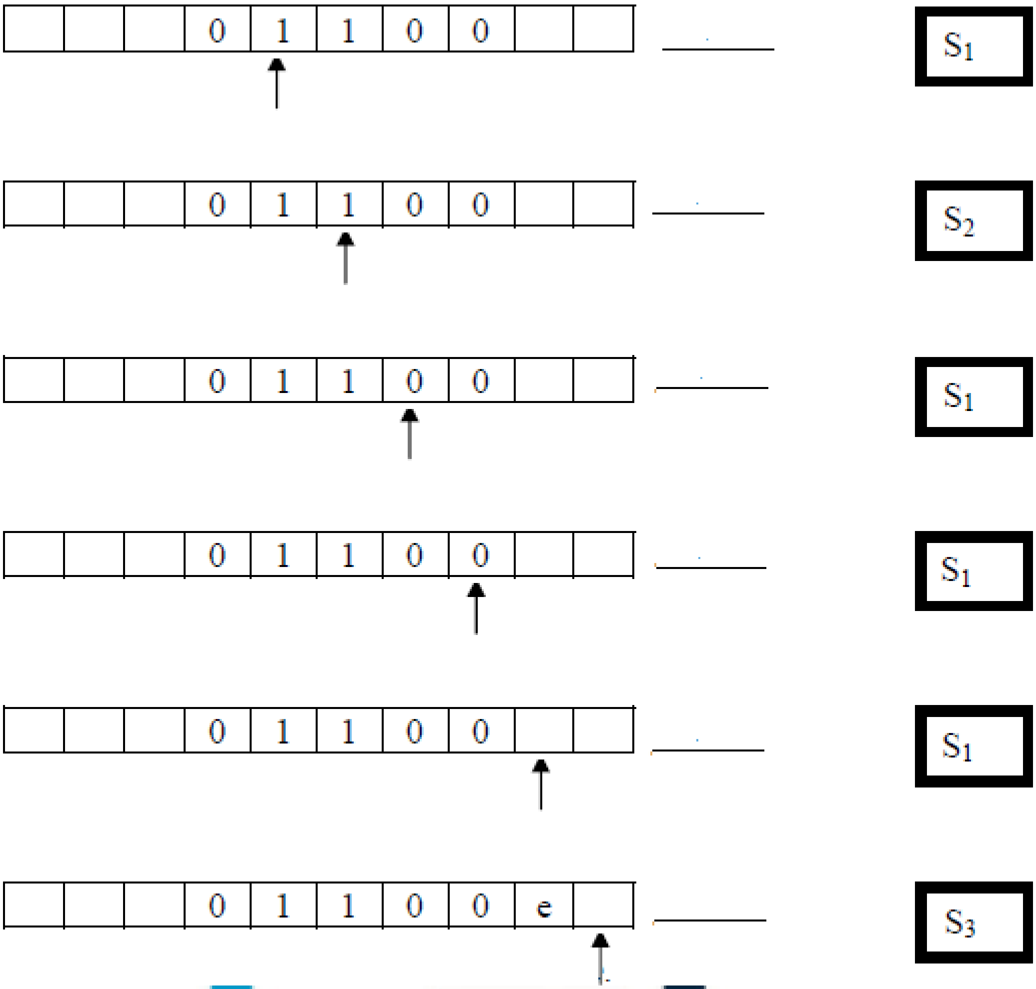
(a)

Number	Correct Label
❶	(0, 0, →)
❷	S_1
❸	(, o, →)
❹	S_3

1 mark for 1 and 3 correct – brackets not required
1 mark for 2 and 4 correct

2

(b)



Mark to end, do not stop at first mistake.
1 mark for first row correct;
1 mark for second row correct;
1 mark for both rows three and four correct;
1 mark for both rows five and six correct;
Must have correct tape contents and current state for each mark

A answers where the tape has been shifted
DPT for missing read/write head

4

(c) Check if the tape contains an even / odd number of 1s // check parity of number on tape;

1

(d) Turing machines provide a (general/formal) model of computation;
Provides a definition of what is computable // a task is computable if (and only if) it can be computed by a Turing machine;
No computing device can be more powerful than a Turing machine // any algorithm that can be computed by any computer can be computed by a Turing machine;
(The Church–Turing thesis states that) if an algorithm exists then there is an equivalent Turing machine for that algorithm // a Turing machine that can implement the algorithm;
Through the Halting Problem, can be used to prove that some functions cannot be computed;

Max 2

[9]

Q1.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen. A number of answers talked about the Halting problem being intractable rather than non-computable.

Q2.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen.

Q3.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen. A Common wrong answer was that a UTM is a Turing Machine that can understand any language.

Q4.

The Halting Problem and Turing Machines were the focus of this question. A number of answers suggested that students had seen a simulation of a Turing Machine but hadn't fully understood what they had seen. A Common wrong answer was that a UTM is a Turing Machine that can understand any language.

Q6.

This question was about Turing machines. After being introduced to the specification as a new topic in 2009 this topic is now fairly well understood.

For part (a) candidates had to consider which aspects of a modern computer system most closely mapped to the purpose of the transition function and tape in the Turing machine, and for part (b) candidates had to explain the relationship between a Turing machine and an algorithm. About a quarter of candidates achieved both marks for (a). For (b), candidates were expected to recognise that if an algorithm existed for a problem then a Turing machine could be created to solve the problem, and vice-versa. It was not enough to just state that a Turing machine could carry out an algorithm; to be markworthy a response had to make clear that a Turing machine could carry out any algorithm.

The trace for part (c) was very well completed with the overwhelming majority of candidates achieving full marks. For part (d) candidates had to explain the purpose of the Turing machine. From just examining the relationship between the input and output, a number of explanations were plausible, but having carried through the process of a trace and having examined the transition function, candidates needed to identify that the Turing machine would reverse the order of the string on the tape. Responses that referred to flipping needed to make clear that it was the order that changed rather than that bits were flipped from 0s to 1s.

Q7.

This question was about models of computation. Question parts (a) and (b) were both extremely well answered with almost all students demonstrating a basic understanding of Finite State Automata by being able to complete the transition table and identify the state correctly.

For part (c) many students were able to identify that there was no way for any string which caused the FSA to enter state S_6 to ever move the FSA to a different state, but to achieve the mark students needed to explain that this was to prevent invalid strings from being accepted, and only around half of students did this successfully. Students sometimes stated that the state would cause the machine to stop or described it incorrectly as a halting state.

Students' understanding of regular expressions has improved significantly during the lifetime of this specification, and it was pleasing to see in part (d) that approximately three quarters of students wrote a fully correct regular expression. Common mistakes were to use the $^+$ operator instead of the * or to misunderstand the scope of the * operator.

For part (e) students were required to explain why a Turing machine was more powerful than an FSA. Good responses recognised that the key difference was the infinite length tape that the Turing machine has, which could be used as an unbounded memory. Whilst an FSA can use its states as a form of memory, this is by definition finite. A commonly seen but incorrect response was that no model of computation could be more powerful than a Turing machine. This explained why a Turing machine was at least as powerful as an FSA, but not why it was more powerful.

Q8.

(a) This part was a trace of the execution of a Turing machine. This was very well tackled with over three quarters of students achieving full marks.

(b) This part required students to explain the overall effect of three of the rules of the Turing machine's transition function. The overall effect was that the tape head would move right along the string until the end of the string was found, without changing the contents of the tape. When the end of the string was located, the state would change to SCO and the head would move left. The most common mistake that students made was to explain what each individual rule did rather than what the purpose of the rules taken together was.

(c) This part was poorly tacked, with only slightly over a quarter of students achieving any marks. A universal Turing machine can be seen to work as an interpreter because it reads instructions in order from a tape and executes them in sequence. This is similar to how an interpreter reads instructions in order from memory and executes them in sequence. Many students either defined what a universal Turing machine was, or answered the question that has been asked on a previous paper about the importance of them.

Q9.

For (a), the vast majority of candidates correctly identified the states and rules from the Turing machine's transition function.

For part (b), as in previous years, the trace of the Turing machine's computation was very well completed. This was particularly pleasing as the transition function was longer and the trace more complex than on previous papers. Approximately three quarters of candidates achieved full marks for this question part.

For part (c)(i), candidates needed to recognise that the x and y symbols were used as placeholders for 0 and 1 so that the Turing machine could identify how much of the string had been processed to achieve a mark. Many fell slightly short of this by recognising the x replaced 0 and y replaced 1, but not explaining the purpose of this replacement.

For part (c)(ii) the majority of candidates explained correctly that that Turing machine could be used to copy binary strings on the tape. Some candidates missed out on the mark by just explaining what the Turning machine had done in this specific execution i.e. “writing 01" to the tape, rather than explaining its more general function. A small number of candidates gave answers taken from previous mark schemes which did not relate to this question.

Q10.

Part (a): This question part was very well answered. The majority of candidates knew how to trace the execution of a Turing machine and many got full marks. The two most common mistakes were to change into state S_2 on the fourth transition, i.e. when the head moved right into the first blank cell, and to start to delete 1s on the fifth transition, i.e. when the head moved left for the first time.

Part (b): The Turing machine deleted the rightmost two 1s from the end of the string on the tape. This was recognised by a third of the candidates. Some however made assertions that were too vague to be creditworthy, such as, “deletes two ones from the tape,” or, “erases the string.” An alternative valid answer was that the Turing machine subtracted two from a unary number. A small number of candidates referred to the end of the tape. Such responses were rejected as the tape is infinitely long.

Part (c): A Universal Turing machine (UTM) is a Turing machine that can simulate the behaviour of any other Turing machine. A description of the Turing machine that is being simulated, including the instructions that the machine follows, is written onto the tape of the UTM. The UTM then acts as an interpreter, faithfully executing the operations on the data exactly as the original Turing machine would have. The two most common mistakes that candidates made were to describe an ordinary Turing machine rather than a Universal Turing machine and to state that a UTM would control another Turing machine rather than simulate it.

Q11.

Part (a): The vast majority of candidates scored both marks for this question. Candidates who only scored one mark usually named the states correctly.

Part (b): Most candidates gained some of the four available marks and over half gained all four.
Some chose to keep the position of the tape head fixed and move the tape, rather than vice-versa which was perfectly acceptable.

Part (c): The Turing machine outputted 'e' if the tape contained an even number of ones and ‘o’ if the number of ones was odd. Determining this was a difficult task given the limited trace that candidates were asked to complete. Nevertheless a quarter of candidates were able to do this.
Many who did not get the correct answer had managed to understand that the use of the Turing machine related to evenness but that this was whether the number itself was odd or even. A commonly made mistake was to assume that the output of the Turing machine depended only on whether the last digit read was a 0 or 1.

Part (d): It was pleasing to see that many candidates understood that a problem is computable if and only if it can be computed by a Turing machine. Some went on to make a further point, such as that no computer could be more powerful than a Turing machine, but answers scoring both marks were rare.