Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

An Roinn Oideachais agus Scileanna
Department of Education and Skills

National Workshop 7

Turing machines

Group work

Overview of the Session

Section 1	Introduction to Turing machines
Section 2	Group activity: Turing machines
Section 3	Group work

By the end of this session participants will have:

- developed their understanding of **Turing machines** and understand their significance
- participated in an activity to develop a deeper understanding of how Turing machines operate
- participated in a discussion on **group work**

PDST

Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section 1

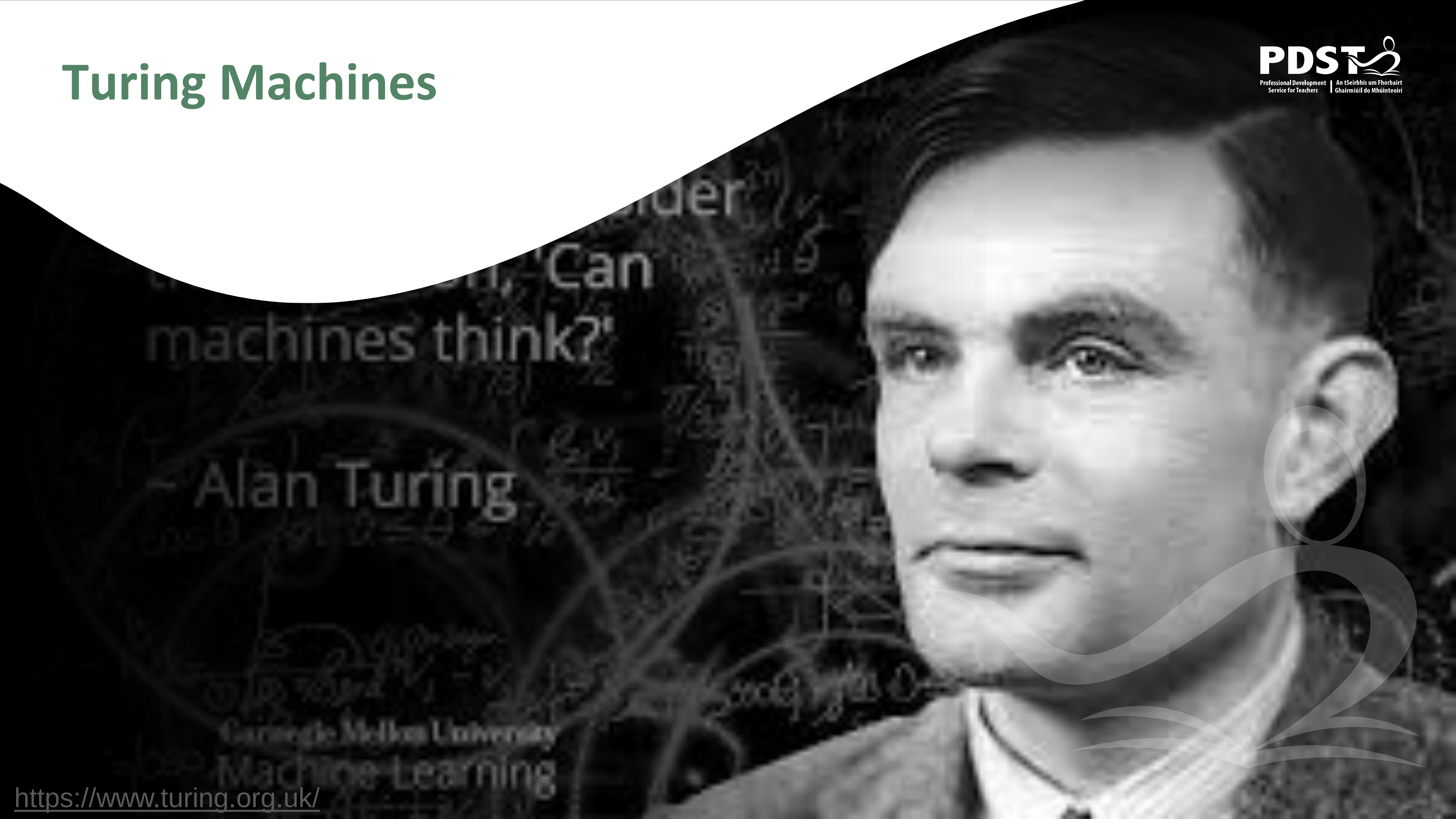
Introduction to Turing machines

We now turn our attention to focus on a fundamental question of Computer Science:

What is computable?

How do we define computability?

Turing Machines



https://www.turing.org.uk/

Turing Machines - Introduction

The illustration below is of an elevator represented as a finite-state machine

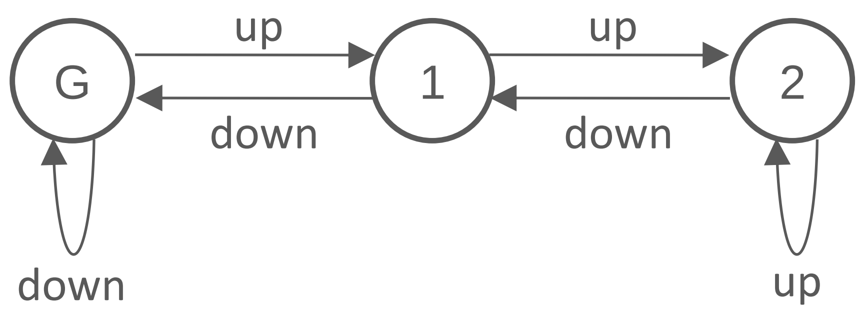
$$\begin{tikzcd}
G \arrow[r, shift left, "up"] \arrow[loop below, "down"] & 1 \arrow[l, shift left, "down"] \arrow[r, shift left, "up"] & 2 \arrow[l, shift left, "down"] \arrow[loop below, "up"]
\end{tikzcd}$$

- Circles represent states (in this case floors)
- Arrows between circles represent transitions between states
- The labels on each transition represents the button press event

What happens when we are on the ground floor and press the **UP** button?
What happens when we are on the ground floor and press the **DOWN** button?

Turing Machines - Introduction

The illustration below is of an elevator represented as a finite-state machine



Assuming we start in state 1, what would the following input sequence yield?

UU DDDD UU D

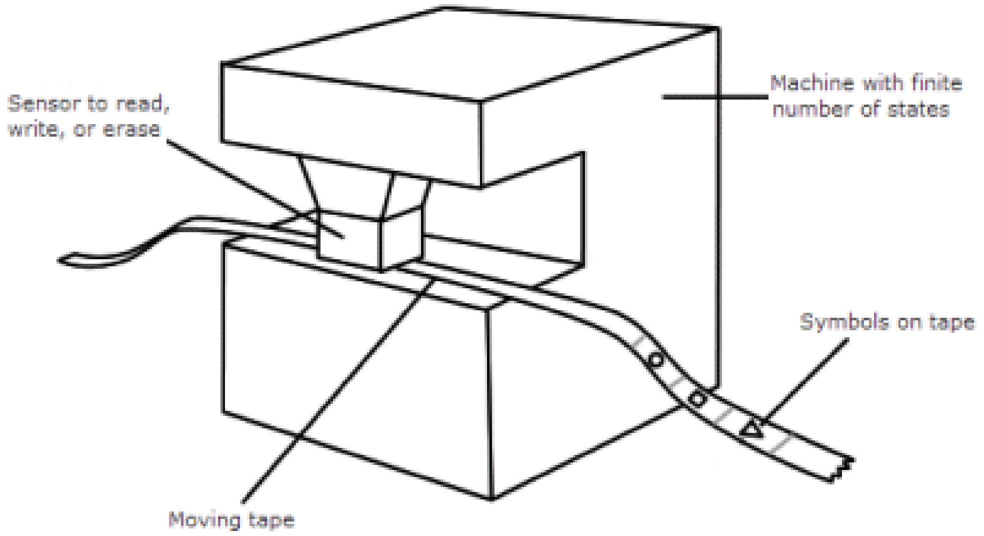
The notion of a Turing machine is not too unlike this…
Given a Finite State Machine and an input, we can determine an output

Turing machines - Introduction

Turing Machines - Introduction

The Turing Machine (TM) was invented in 1936 by Alan Turing.
It is a basic abstract symbol-manipulating device that can be used to simulate the logic of any computer that could possibly be constructed.

Although it was not actually constructed by Turing, its theory yielded many insights.

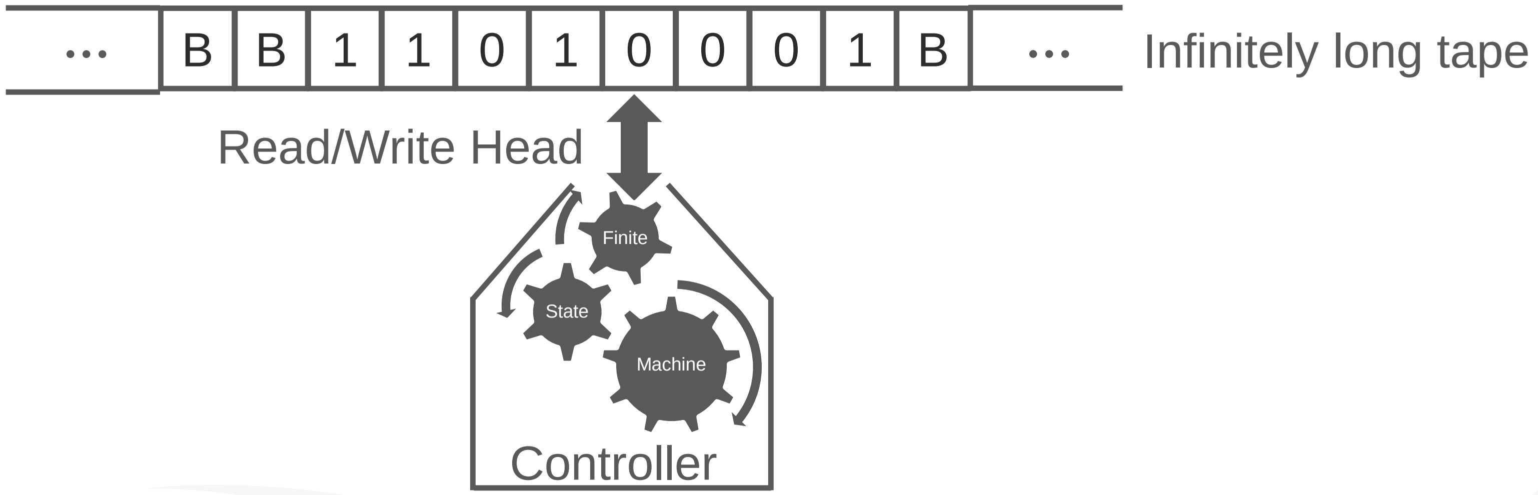

Sensor to read, write, or erase
Machine with finite number of states
Symbols on tape
Moving tape

Anything that is possible to (mathematically) compute could be programmed on a Turing machine.

Turing Machines - Introduction

A Turing Machine consist of three components as follows:

1. **An infinitely long tape made up of individual cells. Each cell can contain a single character – typically 1, 0, or B (blank)**
2. **A read/write head pointed at an individual cell**
3. **A controller (aka finite-state machine) which instructs the read/write head what to do**



A schematic representation of a Turing Machine

Turing Machines - Operation

Initially the tape is inscribed with a sequence of characters – called the input

For example:

…	1	0	1	0	0	1	…

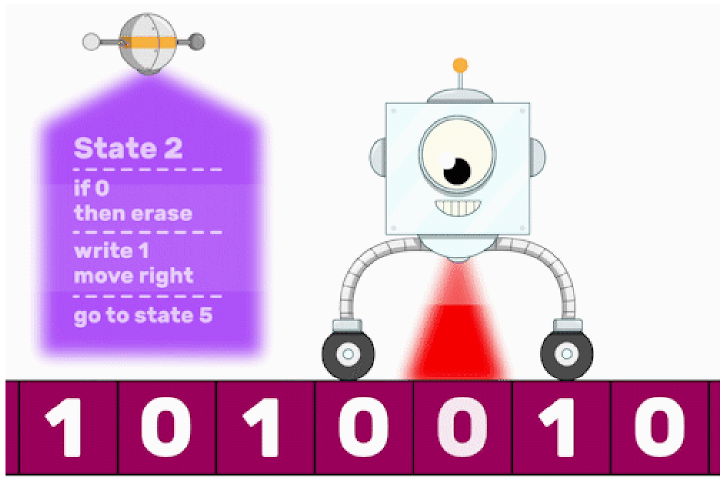
The operation of the Turing Machine is controlled by the finite-state machine (controller).

The operation takes place as a sequence of steps known as transitions

The controller decides for a given (input character, state) pair, the (output character, state) pair - know as a transition.

Each transition involves:

- Reading
- Writing
- Moving
- Updating (state)

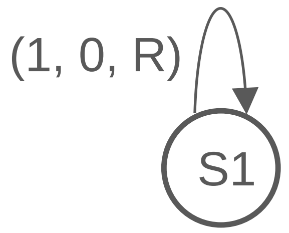


Turing Machines - Operation

Transitions can be expressed using:

state transition tables **OR** **state transition diagrams**

Current State	Read (input)	Write (output)	Direction to move r/w head	Next State
S1	1	0	Right	S1

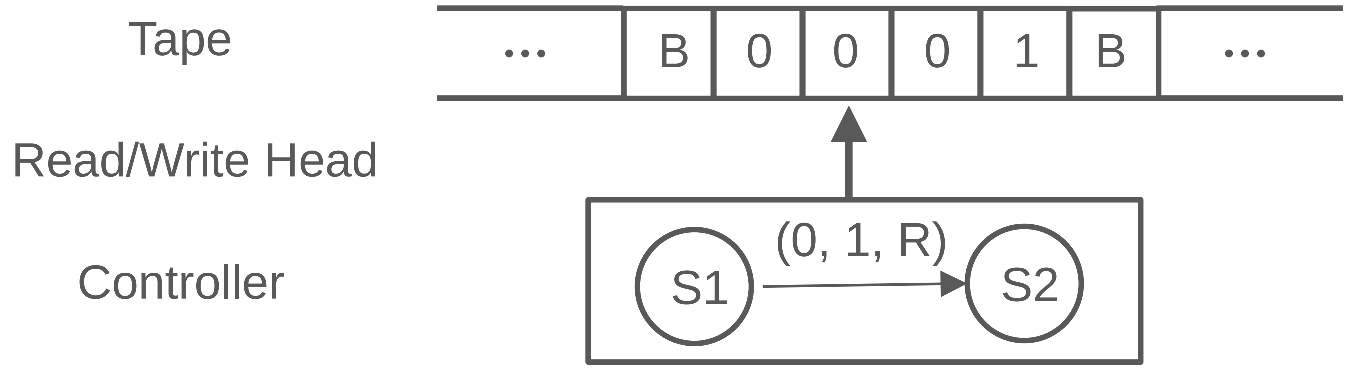


The above state transition table and diagram shows a single transition which says:

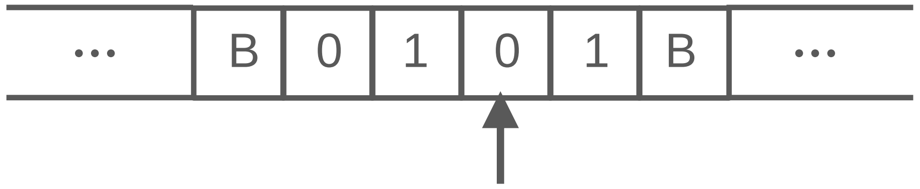
When in state S1 <u>and</u> the symbol being read is a one, write a zero, move right and remain in state S1

Turing Machines - Operation

The illustration below depicts a TM which defines a transition from state S1 to S2 when the current symbol being read in a zero.



After the transition has been completed the symbol zero has been replaced with a 1, the read/write head has been moved right and the new state is set to S2



The result of the computation (output) is the sequence of characters left on the tape if and when the Turing Machine halts.

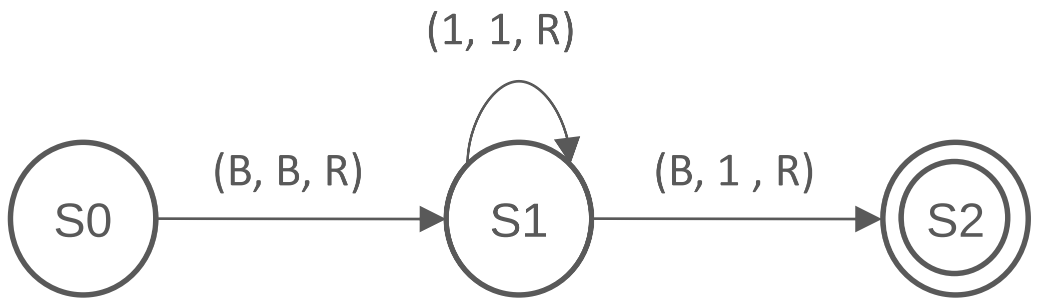
Turing Machines – States

At any given time, a TM is said to be in a particular state. States are usually denoted by the letter S followed by a number e.g. S2 is taken to mean state two.

S0 is conventionally used to denote the initial state. This is the state the TM is in before it starts to operate.

A double circle is used to denote the final or *halting state*. This is the state the TM is in when it finishes.

For example,



Turing Machines – Significance

Earlier we asked the question:
How do we define computability?

Now we can provide the answer:
A task is computable if it can be carried out by a Turing Machine

Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section 2

Group activity: Turing machines

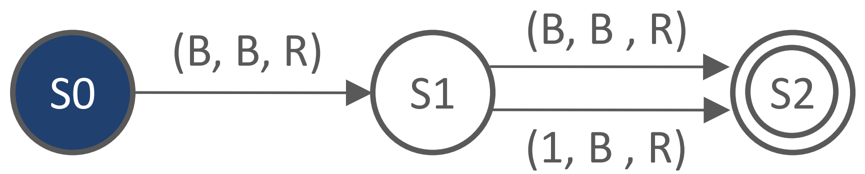
Turing Machine Activity

Each group will trace through the operation of the Turing Machines assigned.

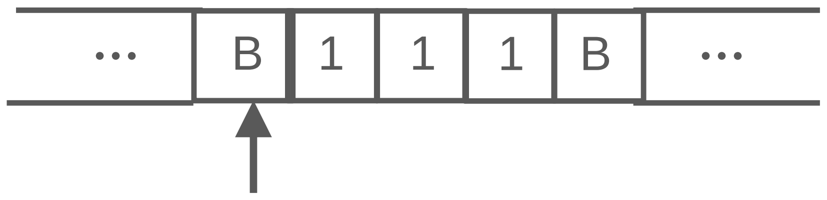
Think
Pair
Share
Square

Turing Machines – Activity – Problem #1

Initial State: S0

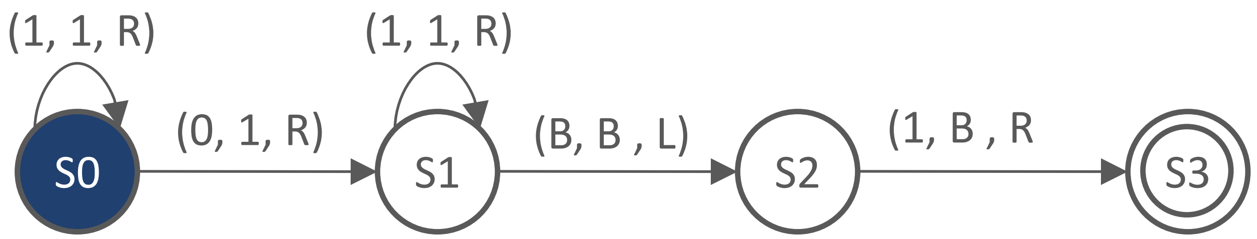


Test input: B111B

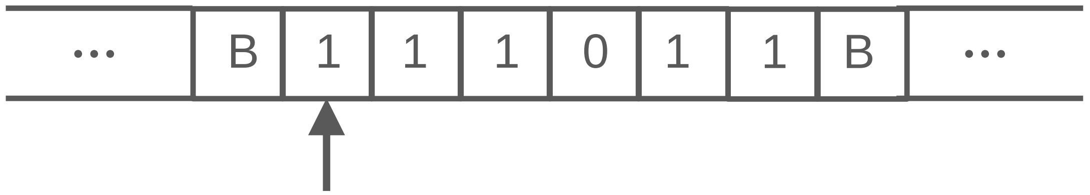


Turing Machines – Activity – Problem #2

Initial State: S0

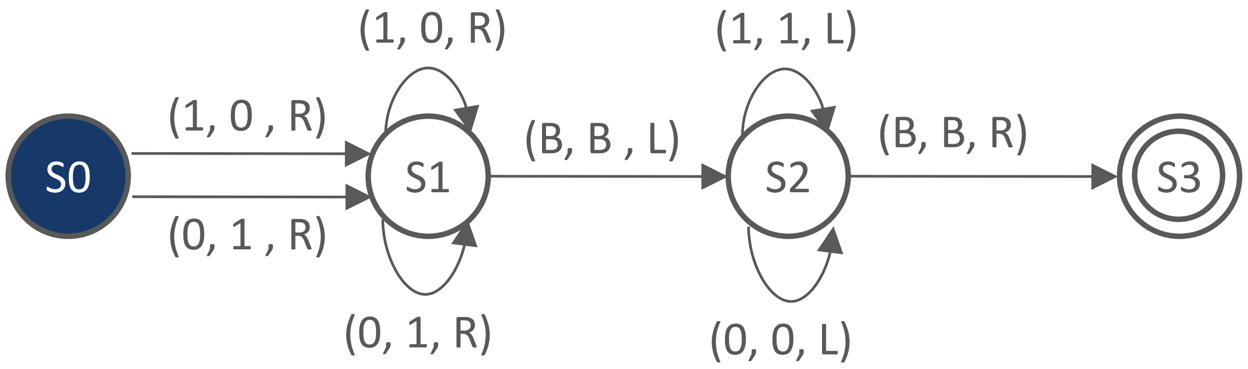


Test input: 111011

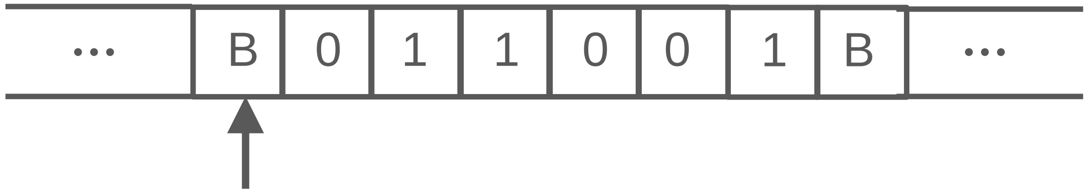


Turing Machines – Activity – Problem #3

Initial State: S0

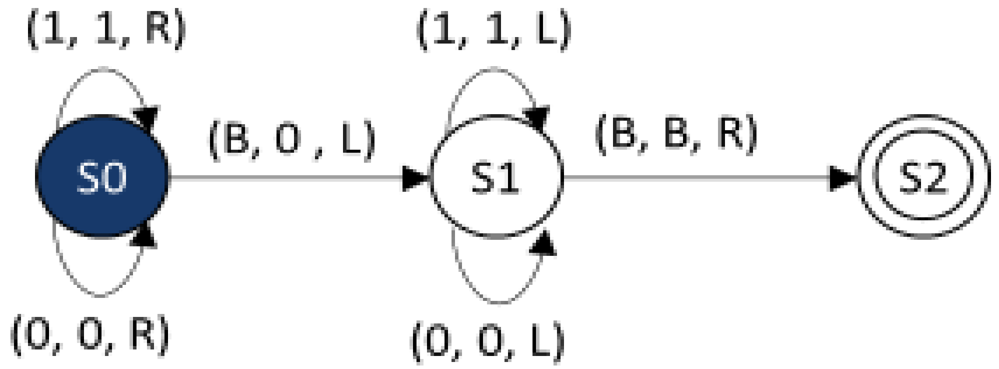


Test input: B011001B

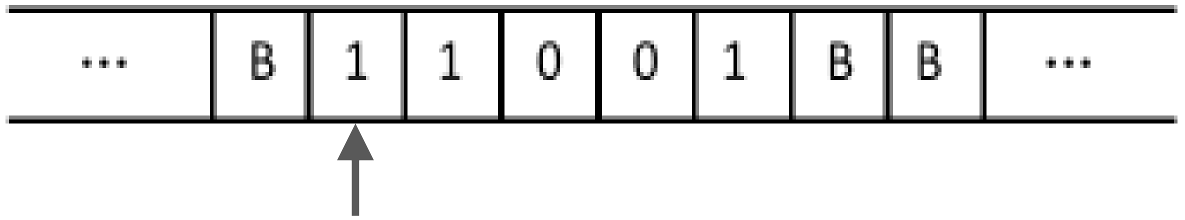


Turing Machines – Activity – Problem #4

Initial State: S0



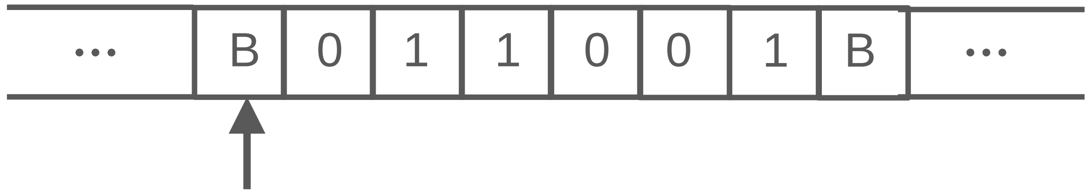
Test input: B11001BB



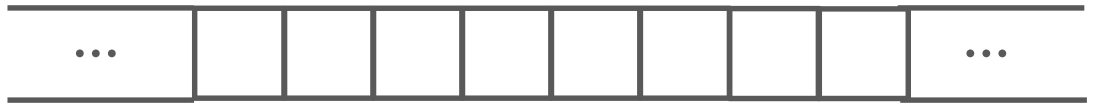
Turing Machines – Activity Handout

Initial Input

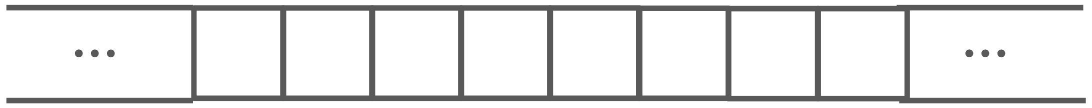
Initial State: S0



After Step 1:
Next State:



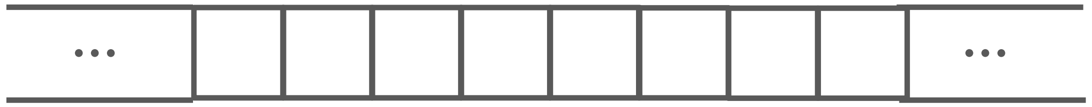
After Step 2:
Next State:



…

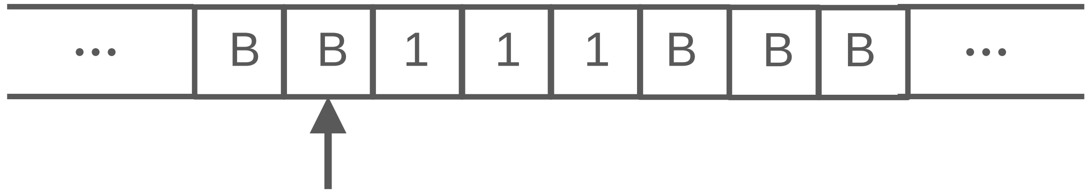
…

Final Output

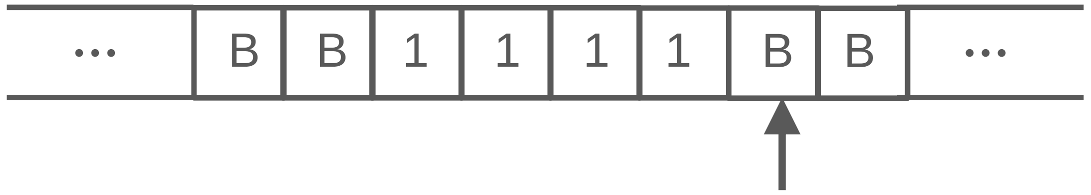


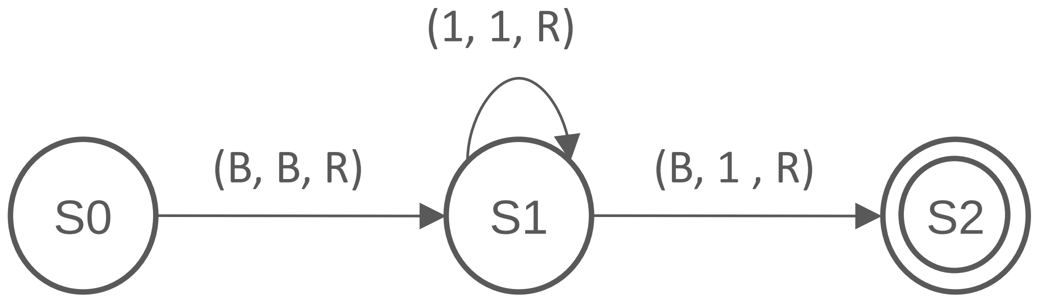
Turing Machines – Example (unary increment)

Test input: 111 (3)



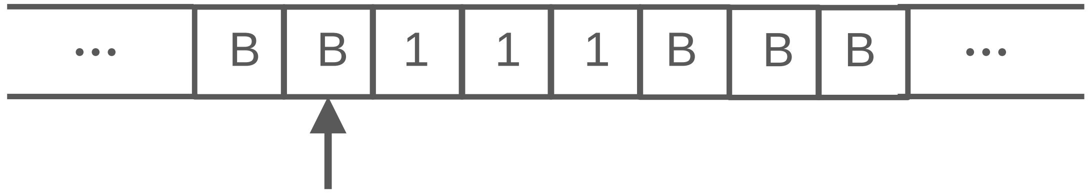
Required output: 1111 (4)



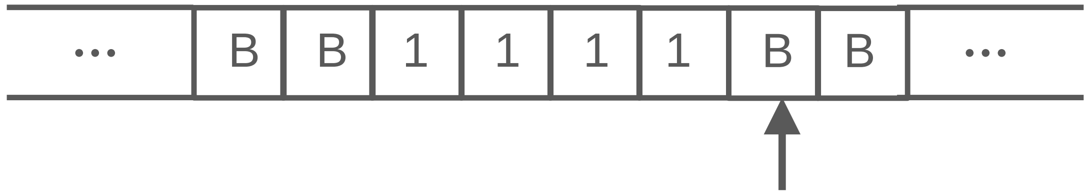


Turing Machines – Example (unary increment)

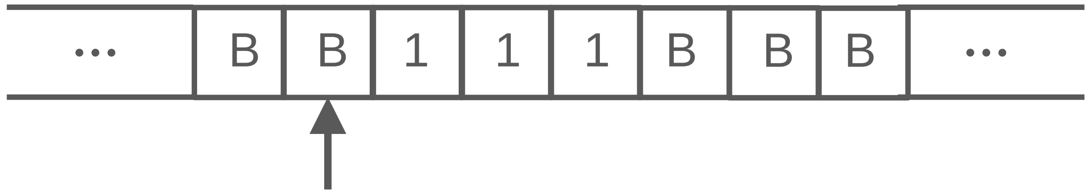
Test input: 111 (3)

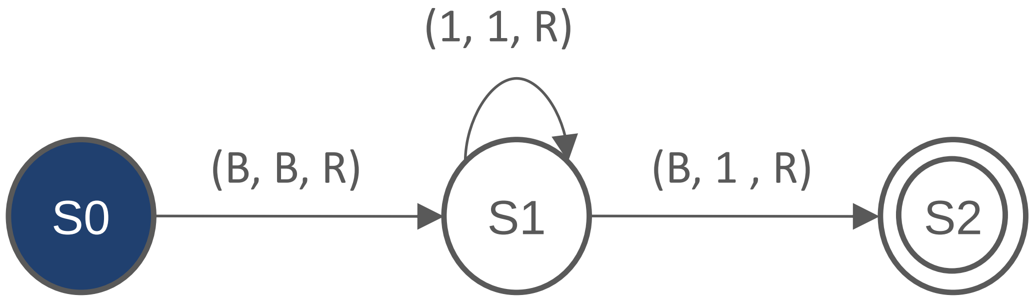


Required output: 1111 (4)



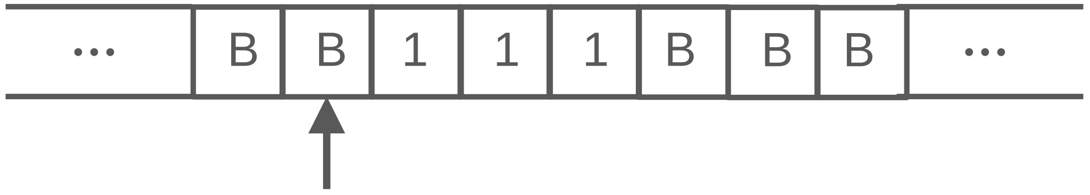
State: S0



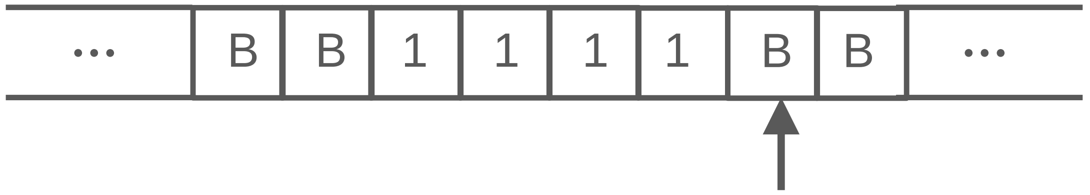


Turing Machines – Example (unary increment)

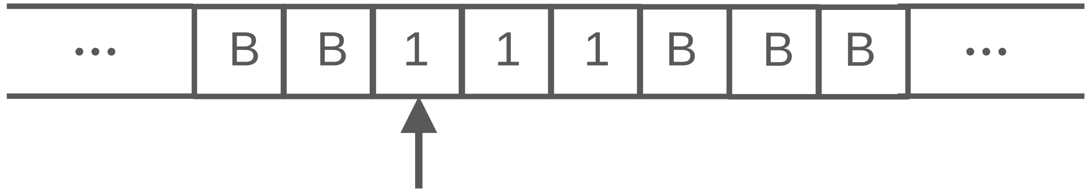
Test input: 111 (3)

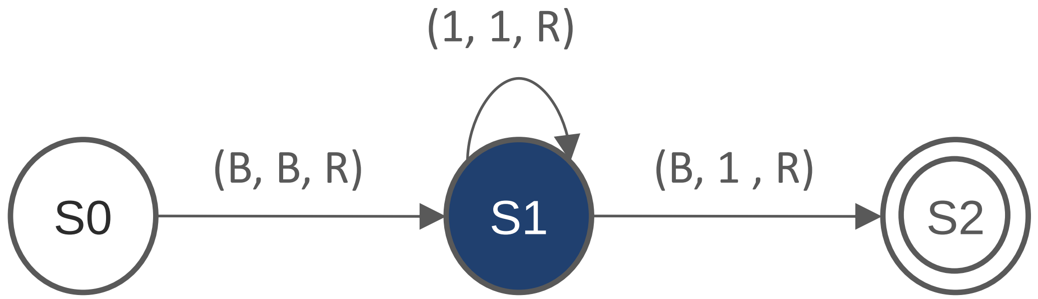


Required output: 1111 (4)



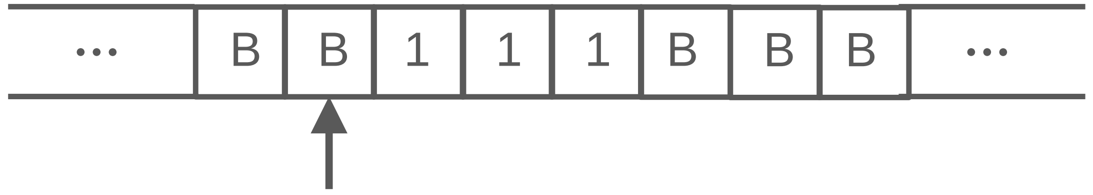
State: S1



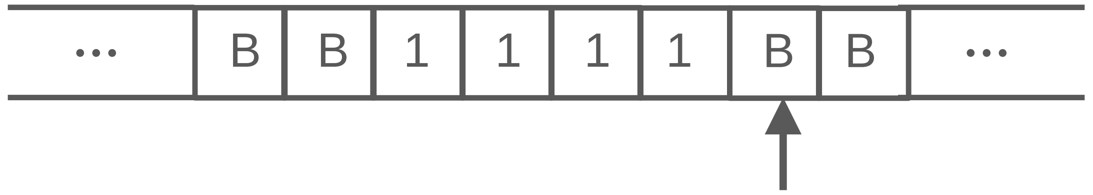


Turing Machines – Example (unary increment)

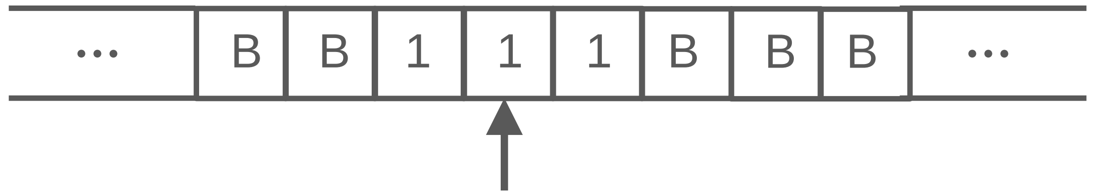
Test input: 111 (3)

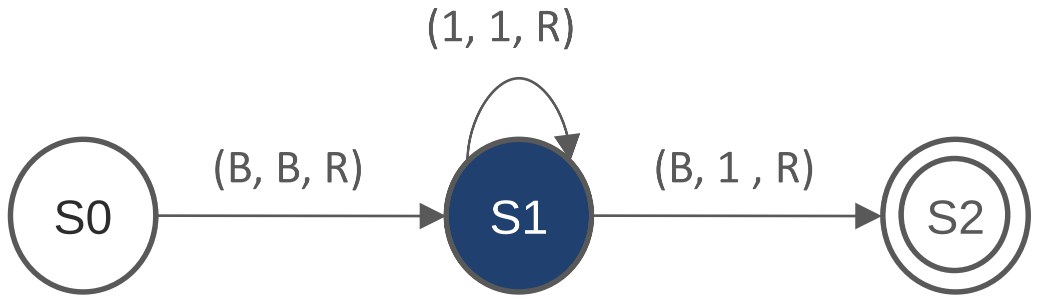


Required output: 1111 (4)



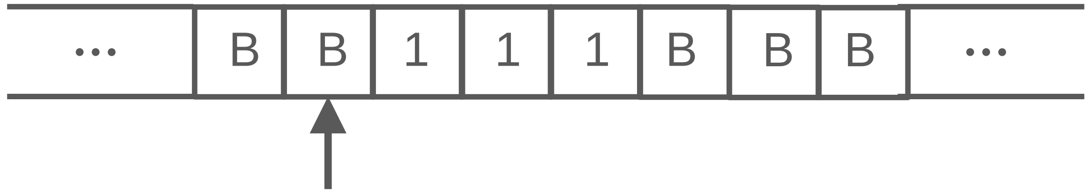
State: S1



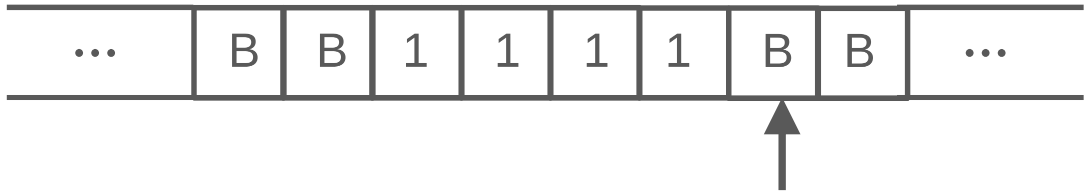


Turing Machines – Example (unary increment)

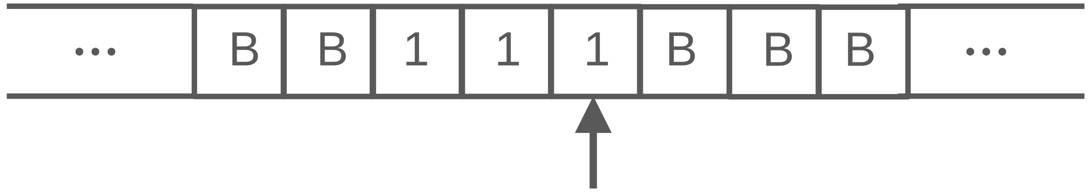
Test input: 111 (3)

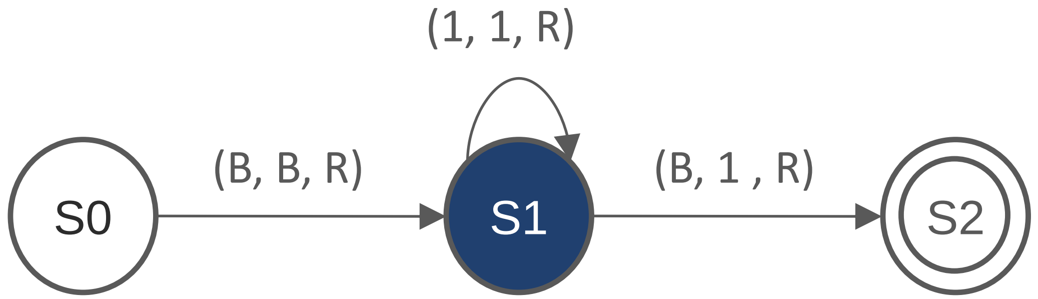


Required output: 1111 (4)



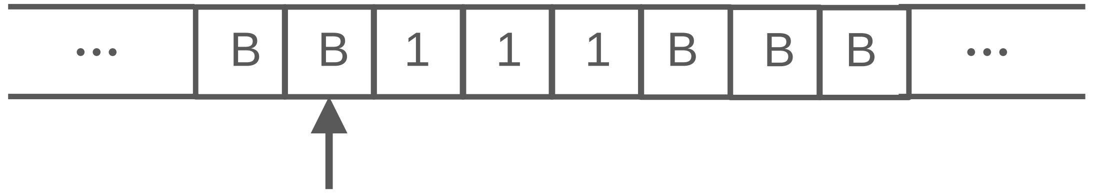
State: S1



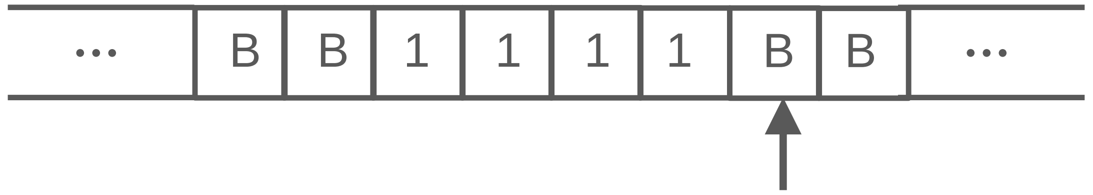


Turing Machines – Example (unary increment)

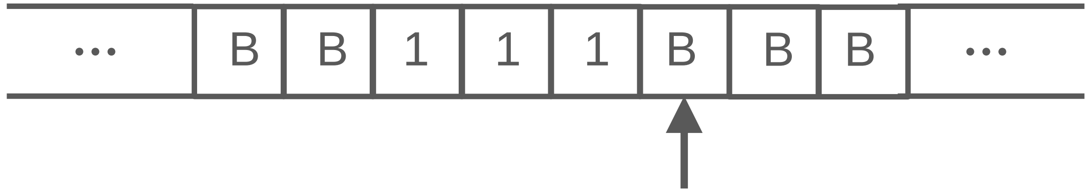
Test input: 111 (3)

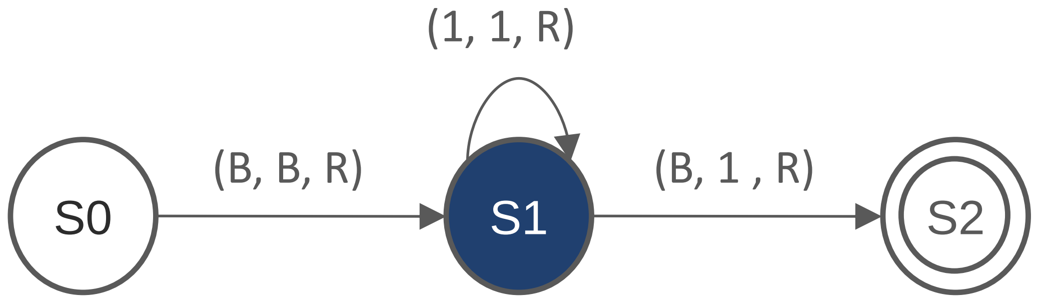


Required output: 1111 (4)



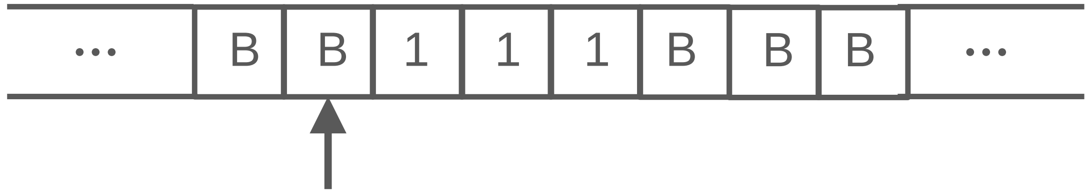
State: S1



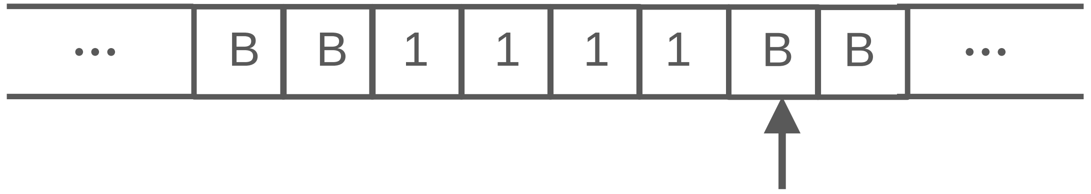


Turing Machines – Example (unary increment)

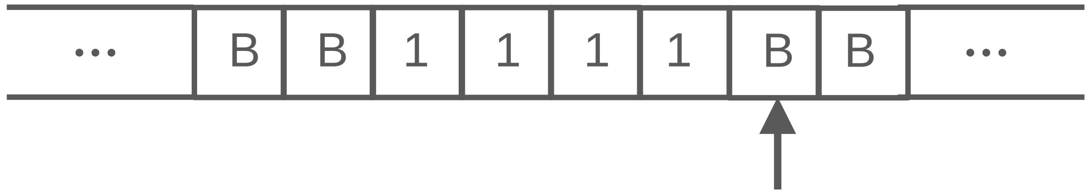
Test input: 111 (3)

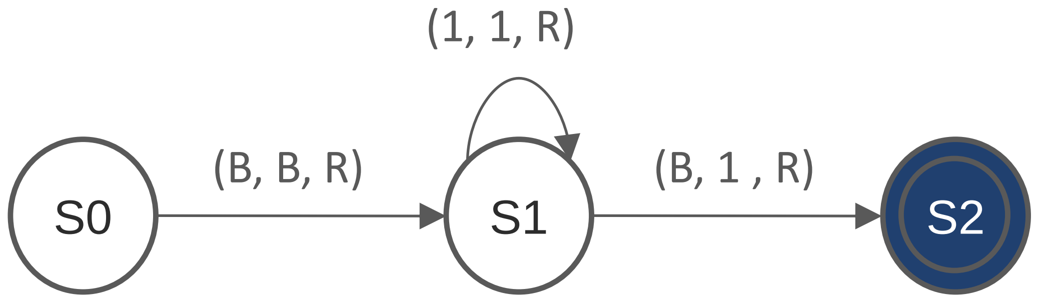


Required output: 1111 (4)



State: S2





Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Section 3

Group work

Group work – Opinions, Tips, Issues

Who sets groups – friends, survey? (RTE)

How many in a group? (Div, pro, act p, coh) (Gross Davis !)

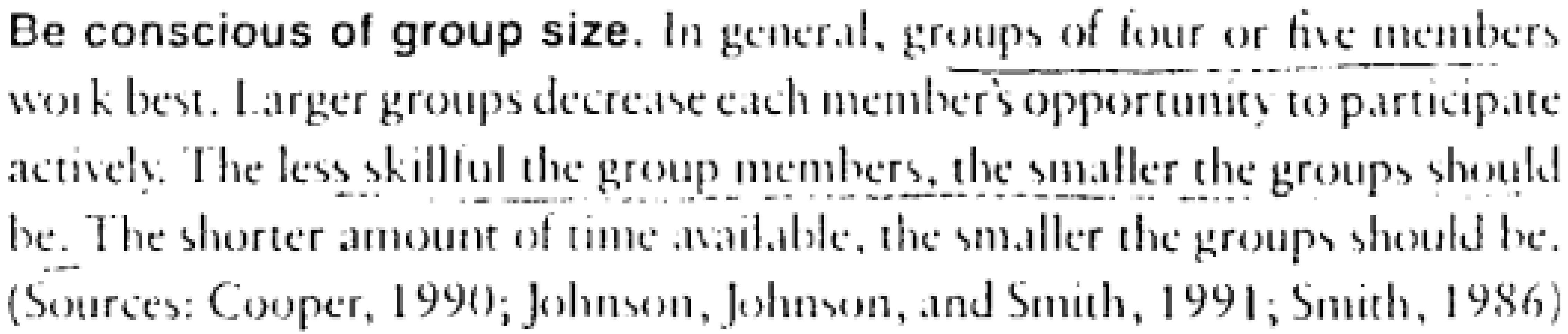
Roles – set by? (Cards)

Challenges?

Individual work? (CS Concepts)

CS - Intermittent Collaboration (Colleen Lewis)

Group Work – Gross Davis



Be conscious of group size. In general, groups of four or five members work best. Larger groups decrease each member's opportunity to participate actively. The less skillful the group members, the smaller the groups should be. The shorter amount of time available, the smaller the groups should be. (Sources: Cooper, 1990; Johnson, Johnson, and Smith, 1991; Smith, 1986)

Professional Development Service for Teachers | An tSeirbhís um Fhorbairt Ghairmiúil do Mhúinteoirí

Exploring group work research

Groupwork – Implementations in Classroom

Implementing Group Work in the Classroom:

https://uwaterloo.ca/centre-for-teaching-excellence/teaching-resources/teaching-tips/alternatives-lecturing/group-work/implementing-group-work-classroom

Group Work articles:

https://drive.google.com/file/d/11kLjGUqF6pQDQoodGairerd1bJHzGqQE/view?usp=sharing

Critical Dialogue

Guidelines for Critical Dialogue

1. Set Container
2. Suspend judgement
3. Listen
4. Listen with Empathy
5. Authentic Voice
6. Opinions based on observations and experiences
7. Allow diversity
8. Listen without Resistance
9. Respect
10. Balance (Advocacy and Inquiry)

An Roinn Oideachais
Department of Education