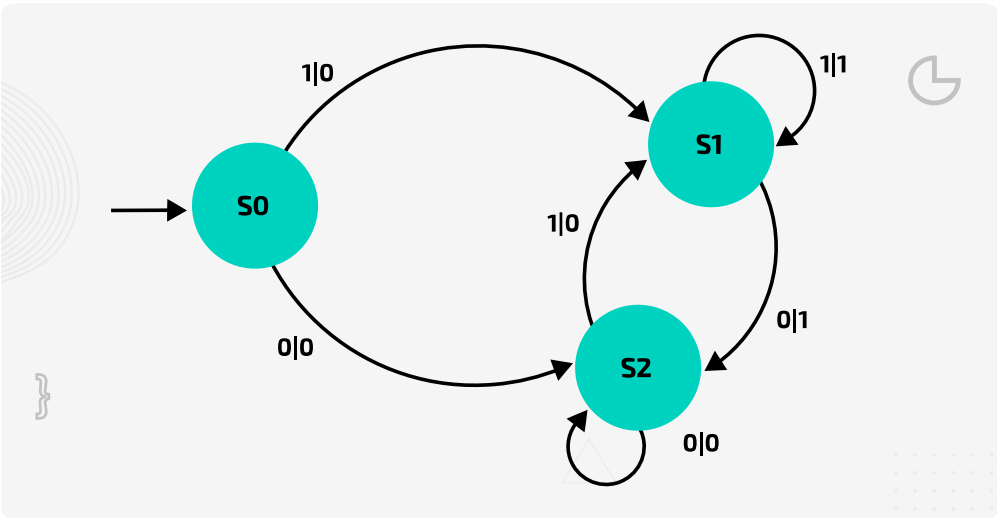


Finite State Machine

Challenge 2



A state transition diagram and state transition table are both methods of representing finite state machines. The diagram and table below represent the same FSM. Identify the missing cell contents in the table.



A finite state machine

Current state	Input	Output	Next state
S0	0	0	S2
S0	1	0	S1
S1	0	1	S2
S1	1	1	S1
S2	0	0	S2
S2	1	0	

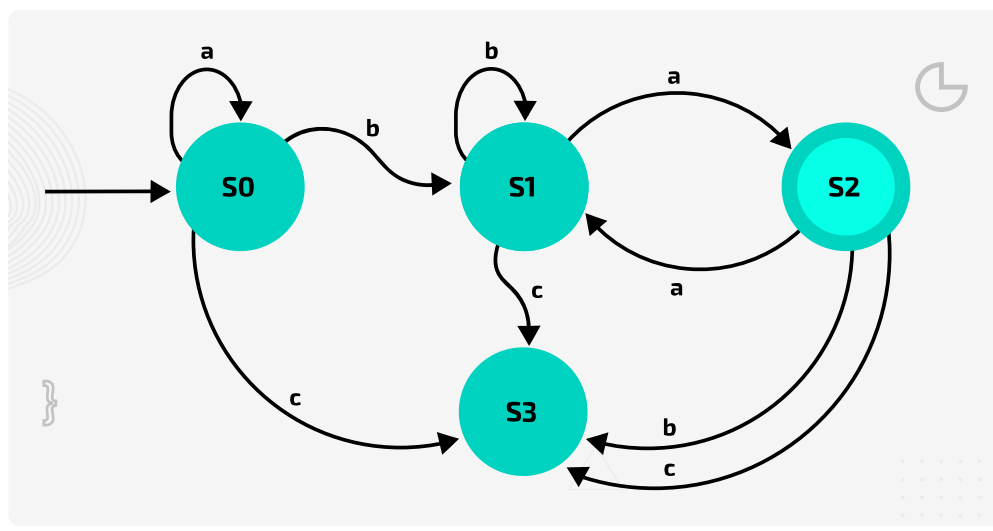
Items:

- S1
- S2
- S0



Strings accepted by FSM

A state transition diagram has been drawn to represent a finite state machine.



A finite state machine

Which **two** of the following strings will be accepted by the FSM?

- ☐ bbaabb
- ☐ aaaaba
- ☐ bbaaba
- ☐ bababa
- ☐ abcabc

Quiz:

STEM SMART Computer Science Week 37

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



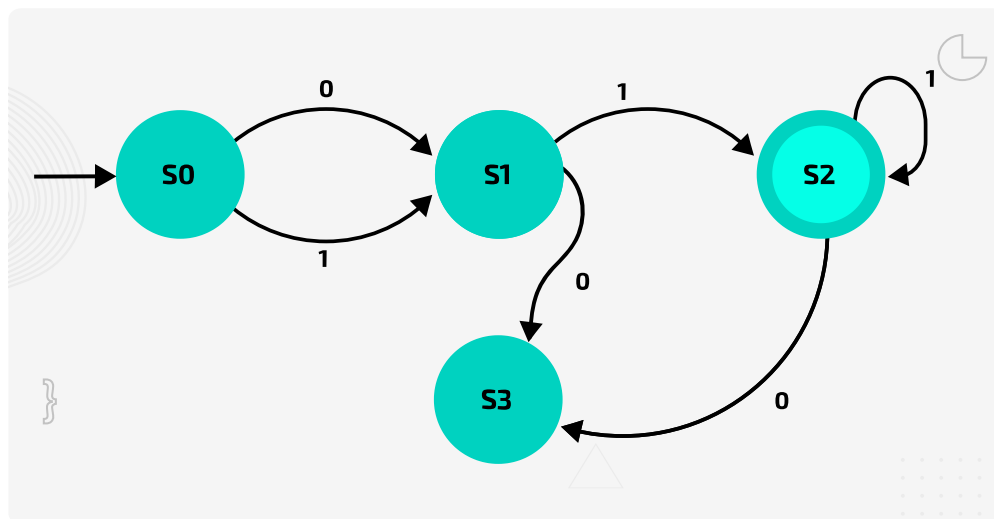
FSM to Regular expression

Challenge 2



A state transition diagram has been drawn to represent a finite state machine

- it has four states
- the initial state is S0
- the accepting state is S2



A finite state machine

Which of the following regular expressions defines the same regular language as that illustrated in the diagram?

- ☐ (0|1)1?
- ☐ 0|11+
- ☐ (0|1)1+
- ☐ (0|1)11+
- ☐ 01?1

Quiz:

STEM SMART Computer Science Week 37



Regular expressions 3

The following statements describe the set of strings matched by the regular expression: **a?(alb)*aba+a** but they are not in the correct order. Put the statements into the correct order (to match the logic of the regular expression).

Available items

zero or one instances of a

one or more instances of a

a single character a

zero or more instances of either a or b

a followed by b

Quiz:

STEM SMART Computer Science Week 37

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



Regular expressions 4

Practice 2



Write a regular expression that will match any string that starts with either a 0 or a 1, followed by zero or more occurrences of the digit 1 (i.e. the expression should match the strings 0, 1, 01, 11, 011, and so on).

Quiz:

[STEM SMART Computer Science Week 37](#)

All teaching materials on this site are available under a [CC BY-NC-SA 4.0](#) license, except where otherwise stated.



Regular expressions 1

Practice 2



The following regular expression has been written to define a valid string:

101*|11?

Which **three** of the following strings will be accepted as valid?

- ☐ 10
- ☐ 011111
- ☐ 11
- ☐ 101101
- ☐ 111111
- ☐ 101111

Quiz:

STEM SMART Computer Science Week 37

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.





Regular language description

Drag and drop the missing words to complete the description of a regular language and finite state machine.

A language is classified as a language if it can be recognised by a . This type of machine consists of a finite number of with between them based on symbols, and a set of accepting states. If a machine ends in an accepting state after processing the entire input string, the string is by the language.

Items:

transitions

states

finite state machine

accepted

input

regular

Quiz:

STEM SMART Computer Science Week 37

All teaching materials on this site are available under a CC BY-NC-SA 4.0 license, except where otherwise stated.



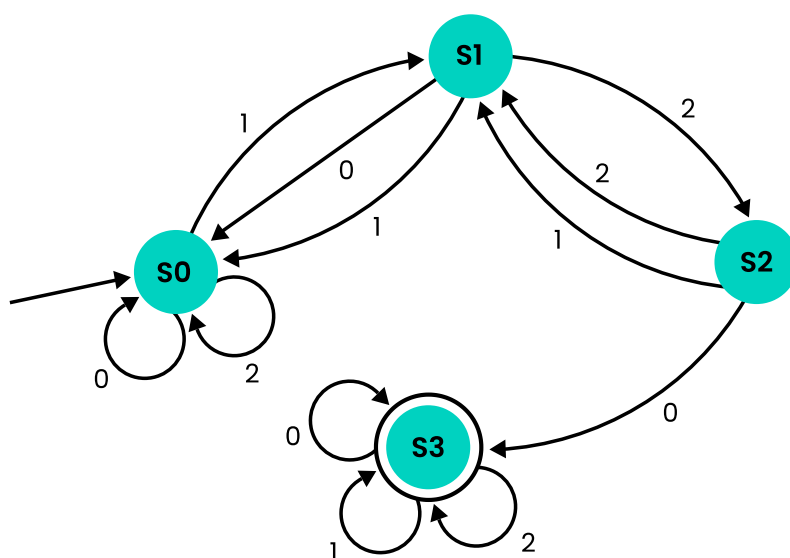
Strings accepted by FSM

Challenge 2



A state transition diagram has been drawn to represent a finite state machine.

- It has 4 states
- The initial state is **S0**
- The accepting state is **S3**
- It accepts strings containing **0,1 and 2**



Select the **3** strings from those listed below that will be accepted by this finite state machine.

- ☐ 111200
- ☐ 1120
- ☐ 2102
- ☐ 1220
- ☐ 120
- ☐ 01200

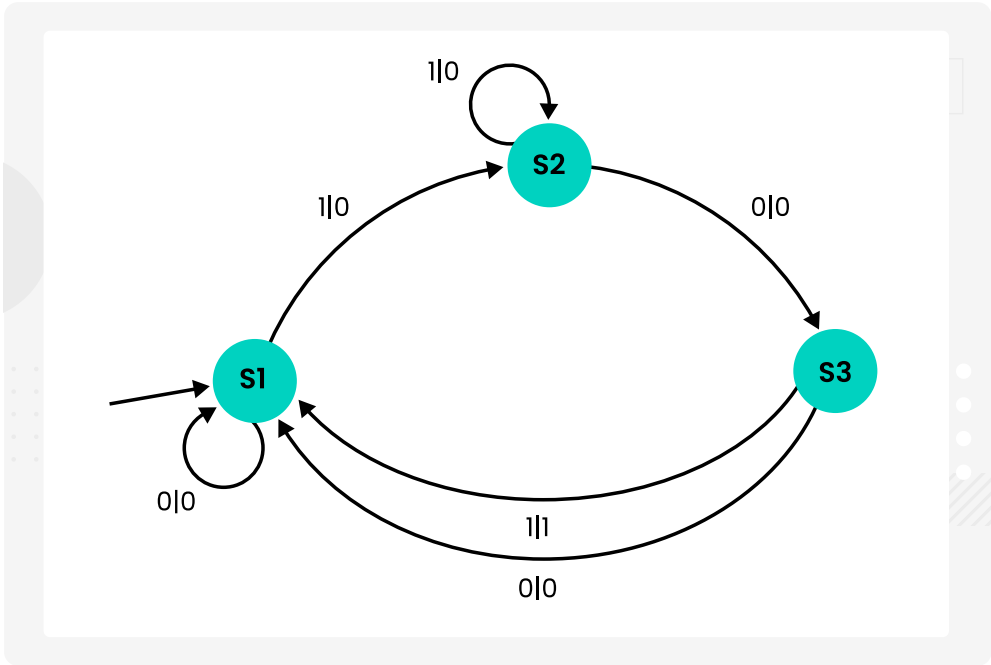
Mealy machine

Challenge 1

A Mealy machine (a finite state machine that produces outputs) is described in the state transition diagram below.

It recognises characters 0 and 1 and consists of 3 states: S1, S2 and S3.

Using the diagram, complete the state transition table below by typing the missing values into the spaces.



Current State	Input	Output	Next State
S1	0	0	S1
S1	1	0	S2
S2	1	0	S2
S2	0	<input type="text"/>	<input type="text"/>
S3	0	<input type="text"/>	<input type="text"/>
S3	1	<input type="text"/>	<input type="text"/>