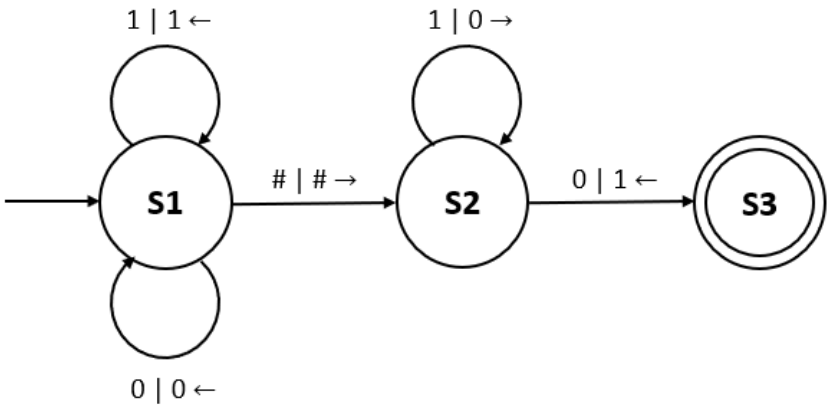
4.4.5 Turing Machines

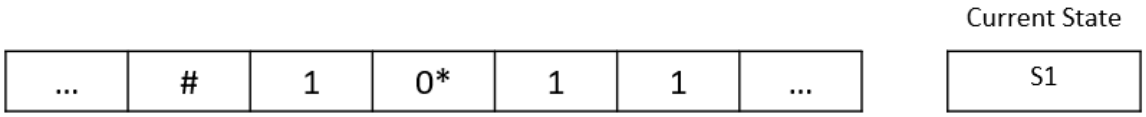
Turing Machines

A Turing machine is a formal model of computation where each machine can be thought of as running a single program. Any computable problem has a Turing machine that can solve it.

A Turing machine consists of:

- A finite set of states
- A set of transition rules
- A finite alphabet of symbols
- An infinite tape
- A read-write head
- A start state
- One or more accept states
- A state register





*A Turing machine with a tape showing #1011 where the * shows the position of the read/write head*

Transition rule

A transition rule specifies:

- the symbol read from the tape
- the symbol to write to the tape
- the direction for the read/write head to move

For example: **0 | 1 →** means if the head reads 0, write 1 and move one square to the right.

State register

Stores the current state of the Turing machine.

Halting state

A state with no outgoing transitions.
An accept state is a halting state that indicates the input has been accepted.

Universal Turing Machine

A Universal Turing machine is a Turing machine that:

- Can execute/simulate the behaviour of any other Turing machine
- It faithfully executes the operations of the simulated Turing machine on data
- The description of the simulated Turing machine and the data are stored on the Universal Turing machines tape.

This is an example of the stored program concept that modern computers are based on. The description of the simulated Turing machine is a program and is stored in memory with the data.

A Universal Turing Machine has an unlimited amount of tape, whereas a real computer has finite memory. This means a Universal Turing Machine is an idealised model rather than a practical computer.