Busy Beaver Lab

Erik Flink, & Dr shakes Chandra

Introduction

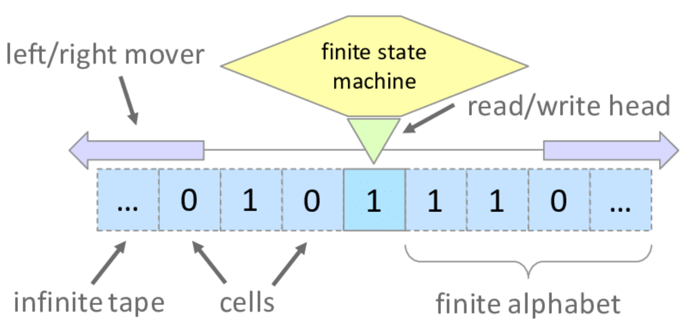
A Turing machine is an abstract mathematical model of a computation machine that can successfully implement any computer algorithm.

Physical Turing machine model

By understanding how a version of a computer in its simplest form runs a range of algorithms, an understanding of how it proved theorems and laid the foundation of computer science can be gained.

Description

The tape head reads then can edit cells according to the state machine and then moves to an adjacent cell, changing states and then repeating this cycle until it reaches the final state or is terminated.



Methods and results

The Turing machines run through a python simulator.

Turing machine simulators

States of simulator 1:

- q0
- saw_#
- saw_##
- qa

States of simulator 2:

- q0
- FindDelimiter0
- FindDelimiter1
- Check0
- Check1
- FindLeftMost
- FindNext
- End
- qa

The machine states can be observed per transition showing how it functions.

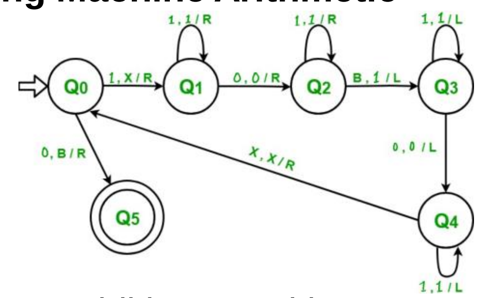
```
q0            [#]#
saw_#         #[#]
saw_##        ##[]
qa            ##[]
```

The first simulator only accepts the input '##'.

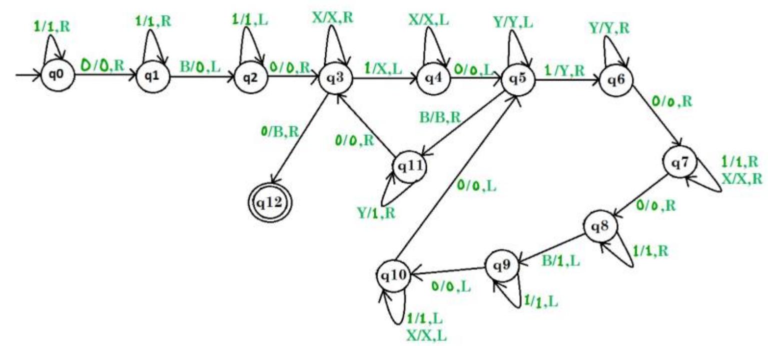
```
q0               [1]0#10
FindDelimiter1   X[0]#10
FindDelimiter1   X0[#]10
Check1           X0#[1]0
FindLeftmost     X0[#]X0
FindLeftmost     X[0]#X0
FindLeftmost     [X]0#X0
FindLeftmost     []X0#X0
FindNext         [X]0#X0
FindNext         X[0]#X0
FindDelimiter0   XX[#]X0
```

The second simulator accepts symmetrical binary strings on both sides of a '#' symbol.

Turing Machine Arithmetic



Addition transition states



Multiplication transition states

The operations inputs and outputs:

2 + 3 = 110111 → XX11111

3 + 4 = 11101111 → XXX1111111

2 × 3 = 110111 → 110XXX111111

3 × 5 = 111011111 → 1110XXXXX111111111111111

Busy Beaver

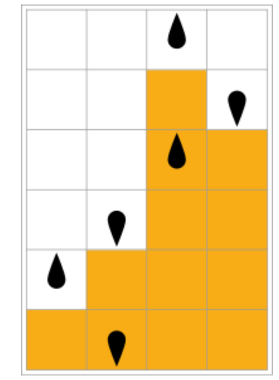
The busy beaver machine aims at finding a halting program that produces the most '1' digits as possible on an infinite blank tape where '' = '0'. Python generators don't hold the list values in memory, instead, it points to one value at a time, simulating a Turing machine.

The number of n-state Turing machines:

$$(4n + 4)^{2n}$$

A	B
0: B1R	0: A1L
1: B1L	1: H1R

2-state cards



Evolution of 2-state

This 2-state busy beaver game produces the most '1' digits possible in 2-states.

Self-produced halting states:

3-card state → 0011000

4-card state → 01011

5-card state → 0111110

As this is only 1 in 24^{10} states for 5 cards it can be assumed that this is unique. The known 5 card produces Σ(5) = 4098 '1' digits. The busy beaver is the fastest-growing function and undecidable.

Conclusion

By simulating different Turing machines, it's clear how diverse their functionality is. As a theoretical model, it proved Alan Turing's mathematical proofs but also allowed modern computers to adapt some of his strategies.