Collatz Conjecture Turing Machine

ian.mcloughlin@atu.ie

March 31, 2023

Input alphabet	$\{0, 1\}$
Tape alphabet	$A \cup \{\#\} \cup \{\_, X\}$
Language	$\{1, 10, 11, 100, 101, 110, 111, 1000, 1001, \ldots\}$
Map	$f(s) \rightarrow 1s\#s$

Duplicate and prepend one

This machine duplicates the input, separating the copy with a comma, and then multiplies the original input by two and adds one to it. We combine this with the machine below, adding the two parts. Together the machines multiply the input, as an integer written in binary with the least-significant bit on the left, by three and adds one.

State	Description
q_0	Append #.
q_1	Move to start.
q_2	If 0/1, mark X go to q_2/q_3.
q_3	Move to end, append 0, go to q_5.
q_4	Move to end, append 1, go to q_6.
q_5	Move X, write 0, back to q_2.
q_6	Move X, write 1, back to q_2.
q_7	Move to start, prepend 1.
q_8	Back to start.

State	Input	Write	Move	Next
q_0	_	#	L	q_1
q_0	0	0	R	q_0
q_0	1	1	R	q_0
q_0	#	#	R	q_f
q_0	X	X	R	q_f
q_1	_	_	R	q_2
q_1	0	0	L	q_1
q_1	1	1	L	q_1
q_1	#	#	R	q_f
q_1	X	X	R	q_f
q_2	_	_	R	q_f
q_2	0	X	R	q_3
q_2	1	X	R	q_4
q_2	#	#	R	q_7
q_2	X	X	R	q_f

q_3	_	0	L	q_5
q_3	0	0	R	q_3
q_3	1	1	R	q_3
q_3	#	#	R	q_3
q_3	X	X	R	q_f
q_4	_	1	L	q_6
q_4	0	0	R	q_4
q_4	1	1	R	q_4
q_4	#	#	R	q_4
q_4	X	X	R	q_f
q_5	_	_	L	q_f
q_5	0	0	L	q_5
q_5	1	1	L	q_5
q_5	#	#	L	q_5
q_5	X	0	R	q_2
q_6	_	_	L	q_f
q_6	0	0	L	q_6
q_6	1	1	L	q_6
q_6	#	#	L	q_6
q_6	X	1	R	q_2
q_7	_	1	L	q_8
q_7	0	0	L	q_7
q_7	1	1	L	q_7
q_7	#	#	L	q_7
q_7	X	X	L	q_7
q_8	_	_	R	q_a
q_8	0	0	R	q_f
q_8	1	1	R	q_f
q_8	#	#	R	q_f
q_8	X	X	R	q_f

Add

This adds two binary numbers, se

State	Description
q_0	Right to #.
q_1	Decrease by 1, go to final states if no 1.
q_2	Left to start.
q_3	Increase by 1, use X as comma if overflow.
q_4	Delete to comma/X, replace X with 1.
q_5	Left to start.

State	Input	Write	Move	Next

q_0	_	_	L	q_f
q_0	0	0	R	q_0
q_0	1	1	R	q_0
q_0	#	#	R	q_1
q_0	X	X	R	q_1
q_1	_	_	L	q_4
q_1	0	1	R	q_1
q_1	1	0	L	q_2
q_1	#	#	L	q_f
q_1	X	X	L	q_f
q_2	_	_	R	q_3
q_2	0	0	L	q_2
q_2	1	1	L	q_2
q_2	#	#	L	q_2
q_2	X	X	L	q_2
q_3	_	_	R	q_f
q_3	0	1	R	q_0
q_3	1	0	R	q_2
q_3	#	X	L	q_0
q_3	X	X	L	q_f
q_4	_	_	L	q_f
q_4	0	_	L	q_4
q_4	1	1	L	q_f
q_4	#	_	L	q_5
q_4	X	1	L	q_5
q_5	_	_	R	q_a
q_5	0	0	L	q_5
q_5	1	1	L	q_5
q_5	#	#	L	q_f
q_5	X	X	L	q_f

Divide by two

Hello

State	Description
q_0	Delete 0, move right.

State	Input	Write	Move	Next
q_0	_	_	L	q_f
q_0	0	_	R	q_0
q_0	1	1	R	q_f
q_0	#	#	R	q_f
q_0	X	X	R	q_f

Remove leading zeros

Fails on string meaning zero.

State	Description
q_0	Move right to end.
q_1	Delete zeros.
q_2	Left to end.

State	Input	Write	Move	Next
q_0	_	_	L	q_1
q_0	0	0	R	q_0
q_0	1	1	R	q_0
q_0	#	#	R	q_f
q_0	X	X	R	q_f
q_1	_	_	R	q_f
q_1	0	_	L	q_0
q_1	1	1	L	q_2
q_1	#	#	R	q_f
q_1	X	X	R	q_f
q_2	_	_	R	q_a
q_2	0	0	R	q_2
q_2	1	1	R	q_2
q_2	#	#	R	q_f
q_2	X	X	R	q_f

Check if 1

State	Description
q_0	Check if first bit 1.
q_1	Check if second bit blank.

State	Input	Write	Move	Next
q_0	_	_	R	q_f
q_0	0	0	R	q_f
q_0	1	1	R	q_1
q_0	#	#	R	q_f
q_0	X	X	R	q_f
q_1	_	_	L	q_a
q_1	0	0	L	q_f
q_1	1	1	L	q_f
q_1	#	#	R	q_f
q_1	X	X	R	q_f

Check if even

State	Description
q_0	Check if first bit 0.

State	Input	Write	Move	Next
q_0	_	_	R	q_f
q_0	0	0	R	q_a
q_0	1	1	R	q_f
q_0	#	#	R	q_f
q_0	X	X	R	q_f

Check if odd

State	Description
q_0	Check if first bit 1.

State	Input	Write	Move	Next
q_0	_	_	R	q_f
q_0	0	0	R	q_f
q_0	1	1	R	q_a
q_0	#	#	R	q_f
q_0	X	X	R	q_f