Tales of the Collatz process in binary

(and ternary) (and senary)

Tristan Stérin, Damien Woods
Hamilton Institute & Computer Science Department
Maynooth University, Ireland

ERC No 772766, SFI 18/ERCS/5746

We present two papers

"Binary representation of ancestors in the Collatz graph",
T. Stérin, RP 2020

"The Collatz process embeds a base conversion algorithm",
T. Stérin and D. Woods, RP 2020

Orthogonal results but same motivation

I. Motivation: why study the Collatz process in binary?
II. Results:
    A. Characterizing the binary representation of ancestors in the Collatz process
    B. The Collatz process can solve base conversion 3 → 2 (not in AC0)

Slides, papers and code: https://dna.hamilton.ie/tsterin/

The Collatz map

- One of the simplest example of *Switched Affine Maps*. Formally introduced in the 60s based on Lothar Collatz's work from the 30s.

J. C. Lagarias. "The 3x+1 problem: an annotated blibliography". 2003, 2006

$$C(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \mod 2 \\ 3x+1 & \text{if } x \equiv 1 \mod 2 \end{cases}$$

$$x \in \mathbb{N}$$

The Collatz map

- One of the simplest example of *Switched Affine Maps*. Formally introduced in the 60s based on Lothar Collatz's work from the 30s.

$$C(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \mod 2 \\ 3x+1 & \text{if } x \equiv 1 \mod 2 \end{cases}$$

Examples: $C(8) = 4 \qquad C(3) = 10$

The Collatz map

- One of the simplest example of *Switched Affine Maps*. Formally introduced in the 60s based on Lothar Collatz's work from the 30s.

$$C(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ 3x+1 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Examples: $C(8) = 4$ $\quad C(3) = 10$

$C(5) = 16$

The Collatz map, slightly fast forwarded

- One of the simplest example of *Switched Affine Maps*. Formally introduced in the 60s based on Lothar Collatz's work from the 30s.

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

$$x \in \mathbb{N}$$

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34,

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34, 17,

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34, 17, 26,

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34, 17, 26, 13,

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let's iterate: 34, 17, 26, 13, 20, 10, 5, 8, 4, 2, **1**

The Collatz process

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let's iterate: 34, 17, 26, 13, 20, 10, 5, 8, 4, 2, **1**, 2, **1**, 2, **1**, ...

The "trivial" cycle.

The Collatz conjecture

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34, 17, 26, 13, 20, 10, 5, 8, 4, 2, **1**, 2, **1**, 2, **1**, ...

Collatz conjecture: All strictly positive integers reach **1** under the action of the Collatz process.

The Collatz conjecture

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

Let’s iterate: 34, 17, 26, 13, 20, 10, 5, 8, 4, 2, **1**, 2, **1**, 2, **1**, ...

Collatz conjecture: All strictly positive integers reach **1** under the action of the Collatz process.

Two seemingly independent components:

1. **Cyclic conjecture:** the only strictly positive cycle is **2,1,2,1,**...
2. **Non-divergence conjecture:** No Collatz trajectory on the strictly positive integers diverges.

As of 2020, **computer tested** up to 2^67

D. Barina. *The Journal of Supercomputing*, 2020.

Is the Collatz problem really about natural numbers?

Is the Collatz problem really about natural numbers?

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \mod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \mod 2 \end{cases}$$

$$x \in \mathbb{N}$$

- $\mathbb{N}(+, \times)$ is a semiring
- There is a "well-defined" **parity function** $\epsilon : \mathbb{N} \to \mathbb{Z}/2\mathbb{Z}$
  - It is an homomorphism: $\epsilon(1) = 1$ $\epsilon(x+y) = \epsilon(x) + \epsilon(y)$
    $\epsilon(x \times y) = \epsilon(x) \times \epsilon(y)$
  - It satisfies:

$$\epsilon(x) = 0 \Leftrightarrow 2 \mid x$$

Is the Collatz problem really about natural numbers?

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

$$x \in \mathbb{Z}$$

Parity function:

$$\epsilon(x) = x \bmod 2$$

Is the Collatz problem really about natural numbers?

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

$$x \in \mathbb{Z}$$

Parity function:
$$\epsilon(x) = x \bmod 2$$

Three known strictly negative cycles:

-1, -1, ...
-5, -7, -10, **-5,** …
-17, -25, -37, -55, -82, -41, -61, -91, -136, -68, -34, **-17,** ...

No known divergent integers

Is the Collatz problem really about natural numbers?

$$T(x) = \begin{cases} x/2 & \text{if } x \equiv 0 \bmod 2 \\ (3x+1)/2 & \text{if } x \equiv 1 \bmod 2 \end{cases}$$

$$x \in \{ \tfrac{a}{b} \mid b \text{ odd } \} \subset \mathbb{Q}$$

Parity function:
$\epsilon(x) = a \bmod 2$

Examples:

-2/7, -1/7, 2/7, 1/7, **5/7**, 11/7, 20/7, 10/7, **5/7**, …

5/3, 3, 5, 8, 4, **2**, 1, **2**, 1, **2**, …

Lagarias' Periodicity Conjecture: J. C. Lagarias. *The American Mathematical Monthly,* 1985.

Every odd-denominator rational reaches a cycle.

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$

$\epsilon(x) = x \bmod 2$

$x \in \mathbb{Z}$

$\epsilon(x) = x \bmod 2$

$x \in \{ \frac{a}{b} \mid b \text{ odd } \} \subset \mathbb{Q}$

$\epsilon(x) = a \bmod 2$

$$\epsilon(x) = 0 \Leftrightarrow 2 \mid x$$

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \{ \tfrac{a}{b} \mid b \text{ odd} \} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$

$$\epsilon(x) = 0 \Leftrightarrow 2 \mid x$$

Binary decomposition algorithm
(i.e., on N, the standard binary encoding)

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \{ \tfrac{a}{b} \mid b \text{ odd} \} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$

$$\epsilon(x) = 0 \Leftrightarrow 2 \mid x$$

while true {
Store $\epsilon(x)$
$x := (x - \epsilon(x))/2$
}

Binary decomposition algorithm (unbounded)

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$

26

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \{\tfrac{a}{b} \mid b \text{ odd}\} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$

while true {
Store $\epsilon(x)$
$x := (x - \epsilon(x))/2$
}

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$

$$x \in \left\{ \tfrac{a}{b} \mid b \text{ odd} \right\} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$

26

$\downarrow$

...0000011010

while true {
    Store $\epsilon(x)$
    $x := (x - \epsilon(x))/2$
}

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$

26

...0000011010

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$

-1

...111111111

$$x \in \{ \tfrac{a}{b} \mid b \text{ odd } \} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$

while true {
    Store $\epsilon(x)$
    $x := (x - \epsilon(x))/2$
}

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$
$\epsilon(x) = x \bmod 2$

26

...0000011010

$x \in \mathbb{Z}$
$\epsilon(x) = x \bmod 2$

-1

...111111111

$x \in \{ \frac{a}{b} \mid b \text{ odd} \} \subset \mathbb{Q}$
$\epsilon(x) = a \bmod 2$

-2/7

...(100)(100)10

while true {
    Store $\epsilon(x)$
    $x := (x - \epsilon(x))/2$
}

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$

$\epsilon(x) = x \bmod 2$

$x \in \mathbb{Z}$

$\epsilon(x) = x \bmod 2$

$x \in \{ \frac{a}{b} \mid b \text{ odd} \} \subset \mathbb{Q}$

$\epsilon(x) = a \bmod 2$

26 → ...0000011010

-1 → ...111111111

-2/7 → ...(100)(100)10

$\mathbb{Z}_2$ The set of “2-adic integers” = semi-infinite binary strings

Is the Collatz problem really about natural numbers?

$$x \in \mathbb{N}$$
$$\epsilon(x) = x \bmod 2$$
26
...0000011010
$$\{(0)^\infty w \mid w \in \{0,1\}^*\}$$

$$x \in \mathbb{Z}$$
$$\epsilon(x) = x \bmod 2$$
-1
...111111111
$$\{(1)^\infty w \mid w \in \{0,1\}^*\}$$

$$x \in \{\tfrac{a}{b} \mid b \text{ odd }\} \subset \mathbb{Q}$$
$$\epsilon(x) = a \bmod 2$$
-2/7
...(100)(100)10
Eventually repeating strings

$$\mathbb{Z}_2$$

"2-adic integers" = the set of semi-infinite binary strings

- $\mathbb{Z}_2(+,\times)$ is a ring (uncountable! unordered!)

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$

$\epsilon(x) = x \bmod 2$

26

...0000011010

$\{(0)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \mathbb{Z}$

$\epsilon(x) = x \bmod 2$

-1

...111111111

$\{(1)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \{\frac{a}{b} \mid b \text{ odd }\} \subset \mathbb{Q}$

$\epsilon(x) = a \bmod 2$

-2/7

...(100)(100)10

Eventually repeating strings

$\mathbb{Z}_2$

“2-adic integers” = the set of semi-inifinite binary strings

- $\mathbb{Z}_2(+, \times)$ is a ring (uncountable!, unordered!)

...11111111
+ …0000001

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$

$\epsilon(x) = x \bmod 2$

26

...0000011010

$\{(0)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \mathbb{Z}$

$\epsilon(x) = x \bmod 2$

-1

...111111111

$\{(1)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \{\frac{a}{b} \mid b \text{ odd}\} \subset \mathbb{Q}$

$\epsilon(x) = a \bmod 2$

-2/7

...(100)(100)10

Eventually repeating strings

$\mathbb{Z}_2$

"2-adic integers" = the set of semi-inifinite binary strings

- $\mathbb{Z}_2(+, \times)$ is a ring (uncountable!, unordered!)

...11111111
\+ …0000001
= …0000000

Is the Collatz problem really about natural numbers?

$x \in \mathbb{N}$

$\epsilon(x) = x \bmod 2$

26

...0000011010

$\{(0)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \mathbb{Z}$

$\epsilon(x) = x \bmod 2$

-1

...111111111

$\{(1)^\infty w \mid w \in \{0,1\}^*\}$

$x \in \{ \frac{a}{b} \mid b \text{ odd } \} \subset \mathbb{Q}$

$\epsilon(x) = a \bmod 2$

-2/7

...(100)(100)10

Eventually repeating strings

$\mathbb{Z}_2$

"2-adic integers" = the set of semi-inifinite binary strings

- $\mathbb{Z}_2(+, \times)$ is a ring (uncountable!, unordered!)
- $\epsilon(x) = \mathrm{LSB}(x)$ is a suitable parity function

$$\epsilon(x) = 0 \Leftrightarrow 2 \mid x$$

Is the Collatz problem really about natural numbers?

$$\mathbb{Z}_2$$

“2-adic integers” = the set of semi-inifinite binary strings

- $\mathbb{Z}_2(+, \times)$ is a ring (uncountable!, unordered!)
- $\epsilon(x) = \mathrm{LSB}(x)$ is a suitable parity function

We can run the Collatz process in $\mathbb{Z}_2$

Lagarias Periodicity Conjecture:

In $\mathbb{Z}_2$, an **eventually periodic** input to the Collatz process yields to an **evenutally periodic** Collatz sequence.

J. C. Lagarias, *The American Mathematical Monthly,* 1985.

Is the Collatz problem really about natural numbers?

Is the Collatz problem really about natural numbers?

No, the Collatz problem is about half-infinite binary strings. (for the scope of this talk)

Is the Collatz problem really about natural numbers?

No, the Collatz problem is about half-infinite binary strings. (for the scope of this talk)

In that context, it is *crucial* to understand the action of the Collatz process in binary.

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory **Computer Science** **Logic**

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory **Computer Science** **Logic**

- By its formulation
- The approach led to numerous results:

J. C. Lagarias. “The 3x+1 problem: an annotated blibliography”. 2003, 2006

- ❏ No strictly positive cycle of length < 17.026.679.261
  S. Eliahou. *Discrete Mathematics*, 1993.

- ❏ “Almost all strictly positive integers almost lead to 1”
  T. Tao. *preprint*, 2019.

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

J. C. Lagarias. "The 3x+1 problem: an annotated blibliography". 2003, 2006

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down):

J.H Conway. *Number Theory Conference*, 1972.

$$G(x) = \begin{cases} x/5+2 & \text{if } x \equiv 0 \bmod 5 \\ 7(x-1)/5+3 & \text{if } x \equiv 1 \bmod 5 \\ 2(x-2)/5+1 & \text{if } x \equiv 2 \bmod 5 \\ 6(x-3)/5+4 & \text{if } x \equiv 3 \bmod 5 \\ (x-4)/5 & \text{if } x \equiv 4 \bmod 5 \end{cases}$$

P. Koiran and C. Moore. *TCS*, 199.

Logic

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

J. C. Lagarias. “The 3x+1 problem: an annotated blibliography”. 2003, 2006

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down).

J.H Conway. *Number Theory Conference*, 1972.

Logic

- Peano-independence in discrete dynamical systems. Ex: Goodstein sequences.

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

Is the Collatz conjecture true?

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down).

Logic

- Peano-independence in discrete dynamical systems. Ex: Goodstein sequences.

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

Is the Collatz conjecture true?

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down).

Computational power of the Collatz process?

Is it Turing complete?

Logic

- Peano-independence in discrete dynamical systems. Ex: Goodstein sequences.

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

Is the Collatz conjecture true?

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down).

Computational power of the Collatz process?

Is it Turing complete?

Logic

- Peano-independence in discrete dynamical systems. Ex: Goodstein sequences.

Peano or ZFC independence?

Computer Science w.r.t the Collatz problem

Three fields (at least) are concerned by the Collatz problem:

Number Theory

- By its formulation
- The approach led to numerous results.

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down).

Logic

- Peano-independence in discrete dynamical systems. Ex: Goodstein sequences.

Computational power of the Collatz process?

Is it Turing complete?

Computer Science w.r.t the Collatz problem

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Maps are Turing complete (exponential slow down)

- Computational power of the Collatz process?
  Is it Turing complete?
- Complexity of prediction problem?

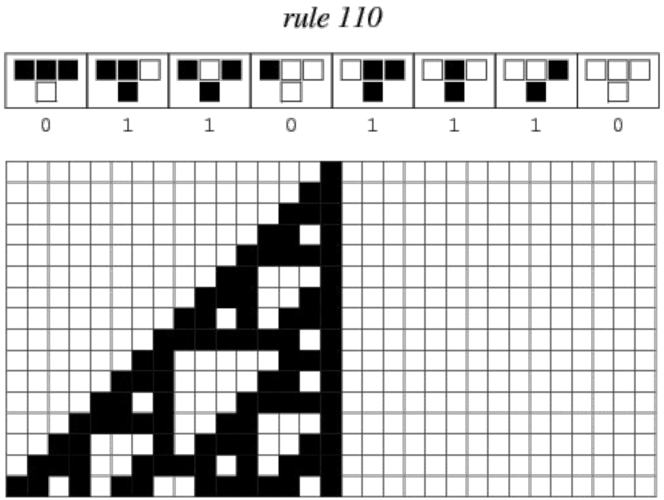
Computer Science w.r.t the Collatz problem

Computer Science

- By its formulation in $\mathbb{Z}_2$
- Generalised Collatz Map e
  Turing complete (exponential own)

- Computational power of the Collatz process?
  Is it Turing complete?
- Complexity of prediction problem?

Very minimalistic discrete dynamical systems are known to be efficient Turing complete models:



M. Cook. *Complex Systems*, 2004

T. Neary and D. Woods. ICALP, 2006.

Motivation

- Computational power of the Collatz process?
  Is it Turing complete?
- Complexity of prediction problem?

The Collatz process in binary

Two operations to understand in binary:

- $x/2$
- $3x+1$

The Collatz process in binary

Two operations to understand in binary:

- x/2
- 3x+1

x/2 corresponds to shifting the binary representation to the right:

...111001010 → ...11100101

The Collatz process in binary

What about 3x + 1 in binary?

$$3x + 1 = x + (2x + 1)$$

Meaning: “x + (**left_shift**(x) + 1)”

Take x = 9 (1001 in binary):

$$\begin{array}{r} 1001 \\ + \quad \end{array}$$

The Collatz process in binary

What about 3x + 1 in binary?

$$3x + 1 = x + (\mathbf{2x} + 1)$$

Meaning: “x + (**left_shift**(x) + 1)”

Take x = 9 (1001 in binary):

$$\begin{array}{r} 1001 \\ + \ \mathbf{10010} \end{array}$$

The Collatz process in binary

What about 3x + 1 in binary?

$$3x + \mathbf{1} = x + (\mathbf{2x} + \mathbf{1})$$

Meaning: “x + (**left_shift**(x) + **1**)”

Take x = 5 (1001 in binary):

$$\begin{array}{r} 1001 \\ + \;\mathbf{10011} \end{array}$$

The Collatz process in binary

What about 3x + 1 in binary?

$$3x + 1 = x + (\mathbf{2x} + 1)$$

Meaning: “x + (**left_shift**(x) + **1**)”

Take x = 9 (1001 in binary):

$$\begin{array}{r} 1001 \\ + \; \mathbf{10011} \\ = \; 11100 \end{array}$$

3*9 + 1 = 28

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$
\begin{array}{r}
1001 \\
+\ 10011 \\
=\ 11100
\end{array}
$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

1001

The Collatz process in binary

$$3x + 1 = x + (2x + 1)$$

$$\begin{array}{rr} & 1001 \\ + & 10011 \\ = & 11100 \end{array}$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

10010

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

  1001
+ 10011
= 11100

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

100010

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

```
   1001
+ 10011
= 11100
```

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$1001\boxed{\bar{0}}$$

$$0$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$\begin{array}{r} 1001 \\ +\ 10011 \\ =\ 11100 \end{array}$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

$$1001\bar{1}\boxed{\bar{0}}$$

$$0$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

```
   1001
+ 10011
= 11100
```

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

10010

0

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$\begin{array}{lr} & 1001 \\ + & 10011 \\ = & 11100 \end{array}$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

10010

00

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$
\begin{array}{rr}
 & 1001 \\
+ & 10011 \\
= & 11100
\end{array}
$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

$$10\bar{0}\boxed{\bar{1}}\bar{0}$$

$$00$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$
\begin{array}{rr}
 & 1001 \\
+ & 10011 \\
= & 11100
\end{array}
$$

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$10\boxed{\overline{0}}\,\overline{1}\,\overline{0}$$

$$00$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

```
   1001
+ 10011
= 11100
```

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$1\,0\,\boxed{\bar{0}}\,\bar{1}\,\bar{0}$$

$$100$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$\begin{array}{r} 1001 \\ +\ 10011 \\ =\ 11100 \end{array}$$

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

$$1\boxed{0}\bar{0}\bar{1}\bar{0}$$

$$100$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$
\begin{array}{r}
1001 \\
+\ 10011 \\
=\ 11100
\end{array}
$$

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$1\boxed{0}\overline{0}\,\overline{1}\,\overline{0}$$

$$1100$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$
\begin{array}{rr}
 & 1001 \\
+ & 10011 \\
= & 11100
\end{array}
$$

$$..00\boxed{1}00\bar{0}1\bar{1}\bar{0}$$

$$1100$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

$$
\begin{array}{r}
1001 \\
+\ 10011 \\
=\ 11100
\end{array}
$$

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$..00\boxed{1}0\overline{0}\overline{1}\overline{0}$$

$$11100$$

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

```
   1001
+ 10011
= 11100
```

Meaning:

"Each bit of the input gets added to its right neighbour and the potential carry on that neighbour."

..0010010

..011100

The Collatz process in binary

$3x + 1 = x + (2x + 1)$

Meaning:

“Each bit of the input gets added to its right neighbour and the potential carry on that neighbour.”

$$
\begin{array}{r}
1001 \\
+ \quad 10011 \\
= \quad \mathbf{11100}
\end{array}
$$

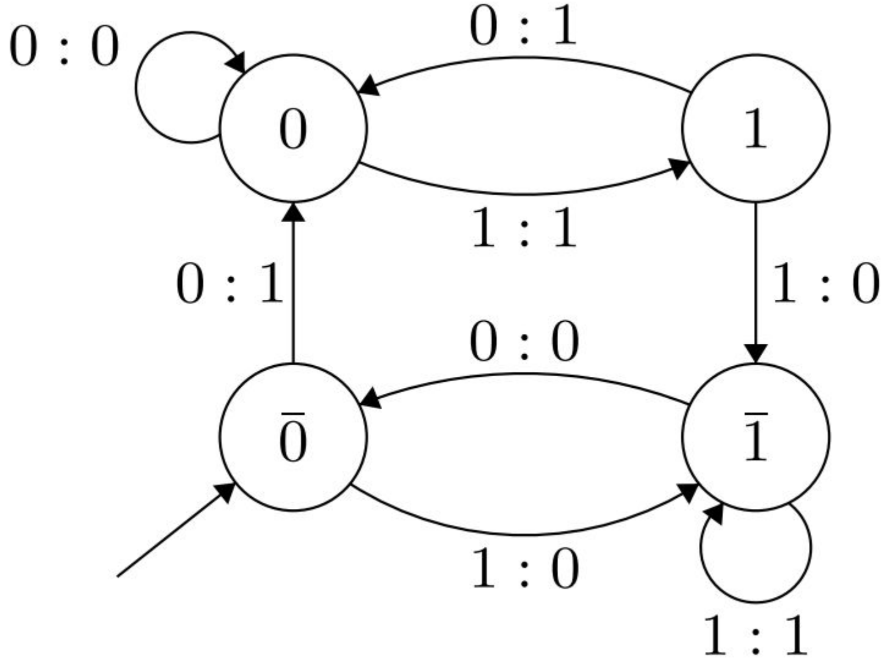
..00100010

..011100

3*9 + 1 = 28

The 3x+1 operation is 2-automatic

Caption
read : write

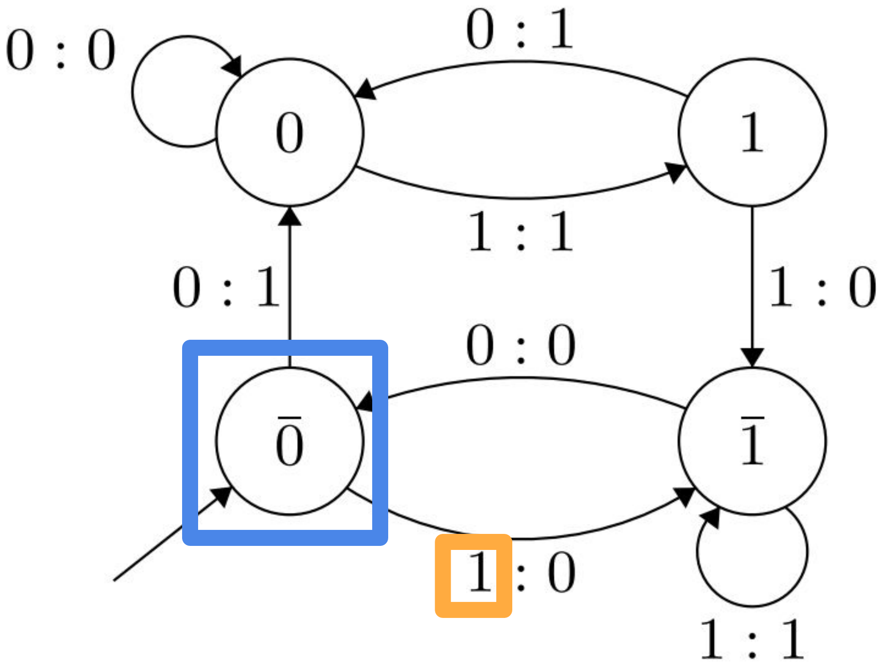


The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

100010



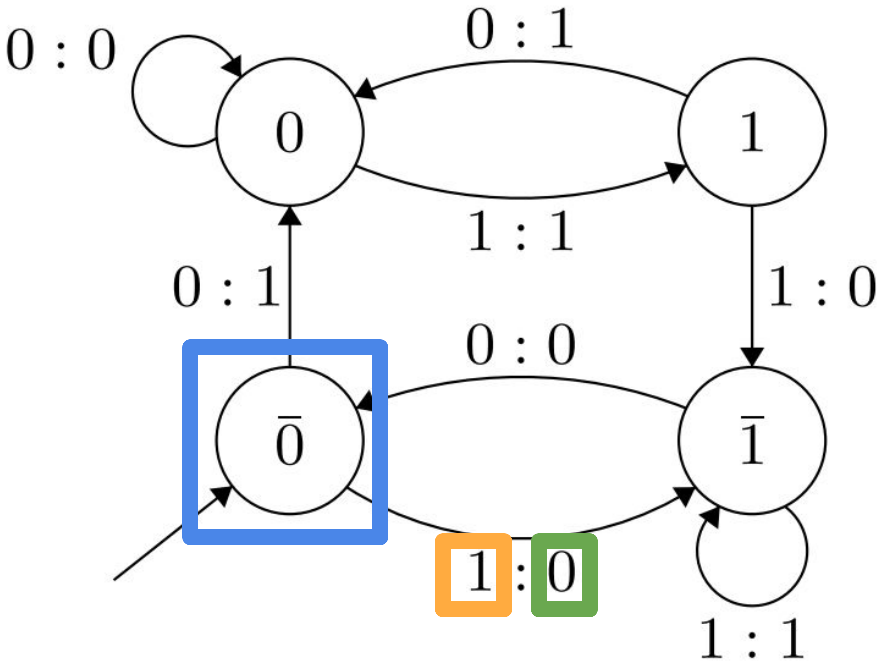
The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

1001$\bar{0}$

0



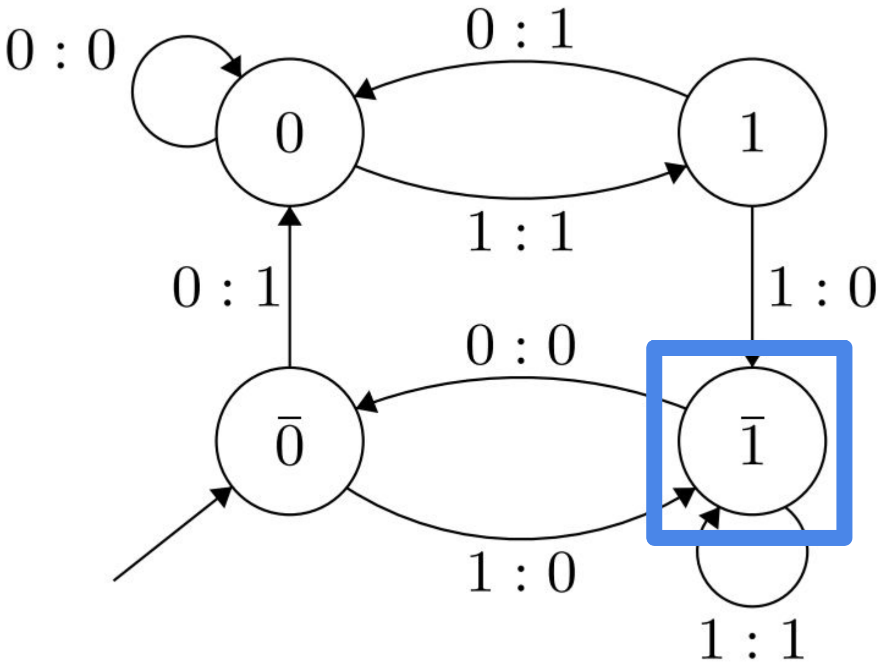
The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

$100\bar{1}\bar{0}$

0



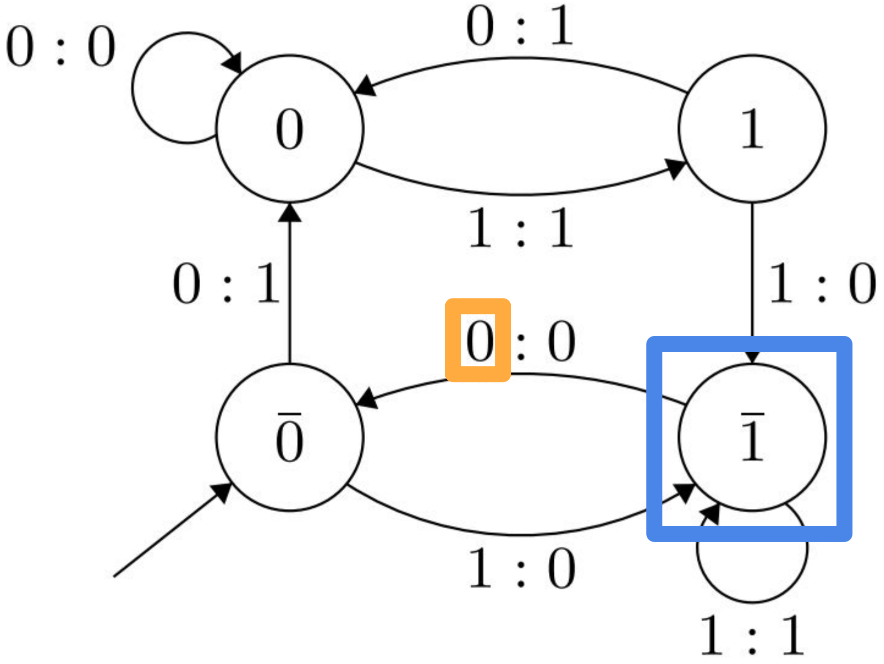
The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

$100\bar{1}\bar{0}$

0



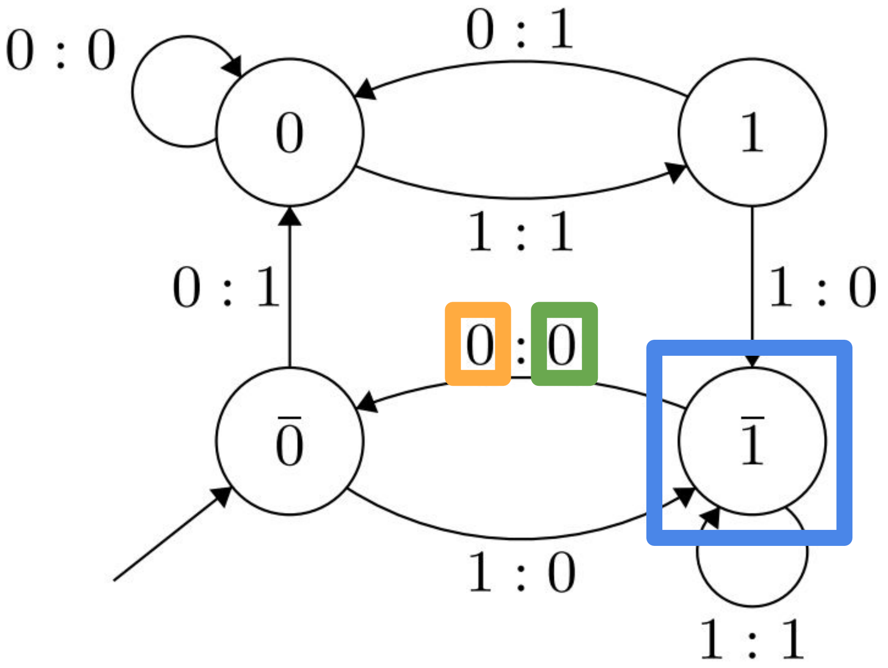
The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

$$100\bar{1}\bar{0}$$

$$00$$



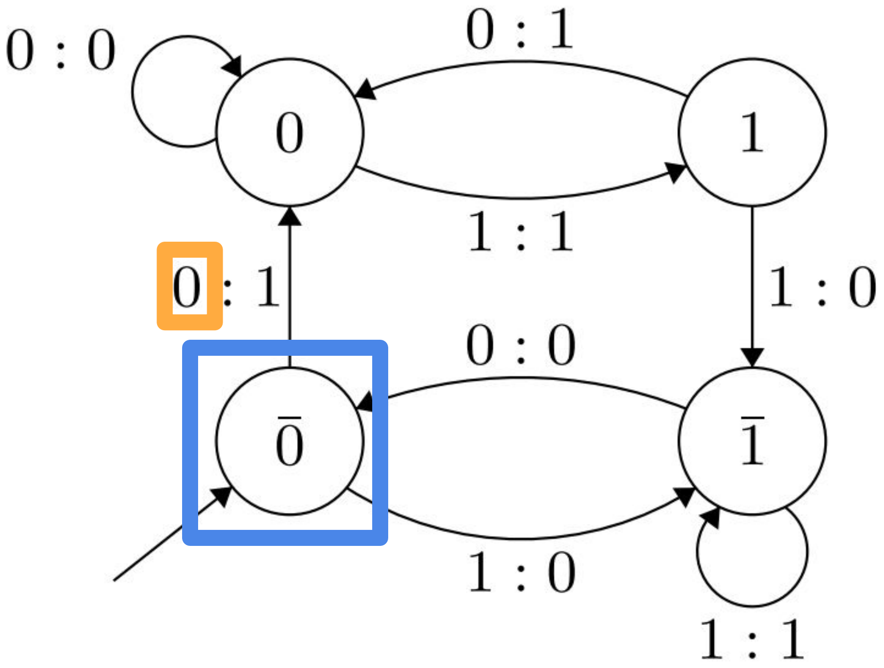
The 3x+1 binary FST

The 3x+1 operation is 2-automatic

Caption
read : write

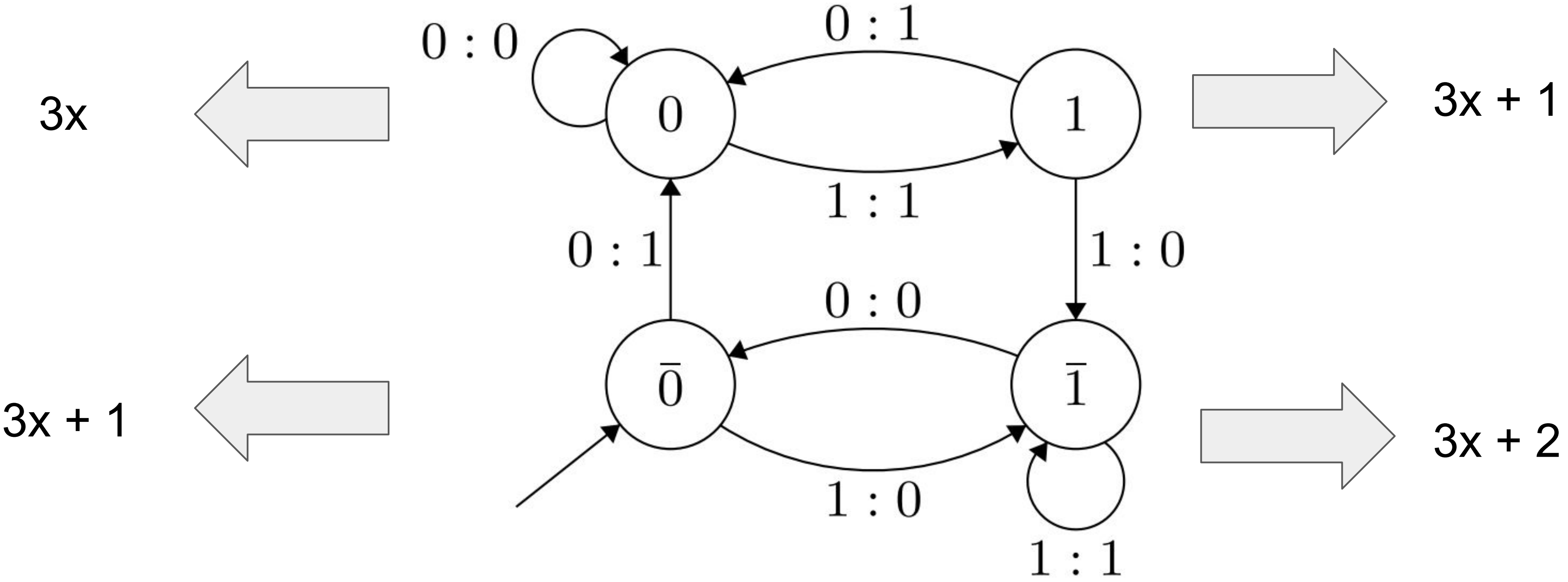
$1\,0\,\bar{0}\,\bar{1}\,\bar{0}$

00

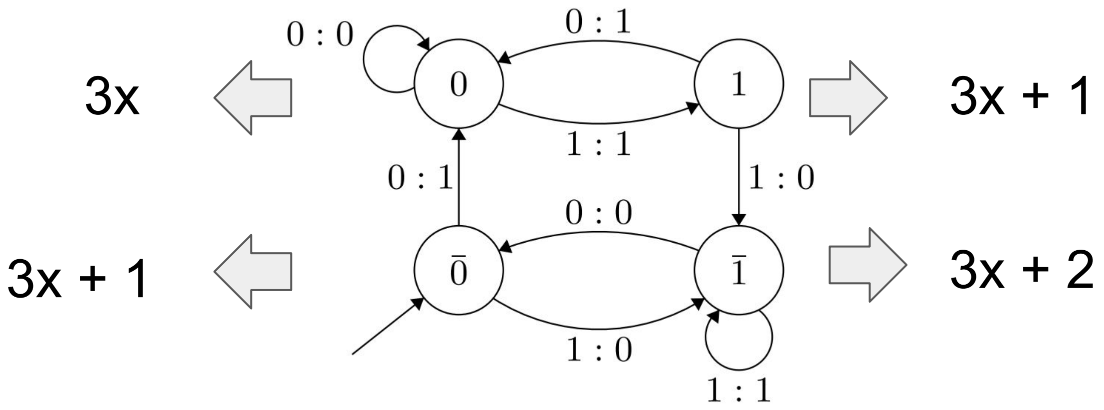


The 3x+1 binary FST

The 3x, 3x+1 and 3x+2 operations are 2-automatic



The 3x, 3x+1 and 3x+2 operations are 2-automatic



“Binary representation of ancestors in the Collatz graph”,
T. Stérin, RP 2020 → Regular expressions

“The Collatz process embeds a base conversion algorithm”,
T. Stérin and D. Woods, RP 2020 → Base 2 → Base 3

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Let’s run the Collatz process on 642:

642 1010000010

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Let’s run the Collatz process on 642:

321 1010000010

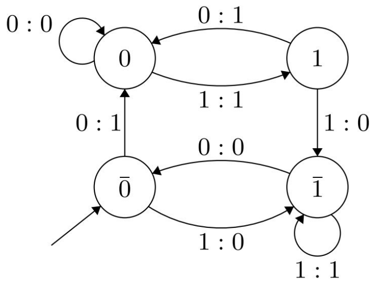
Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Let’s run the Collatz process on 642:

321   1010000010

964   1111000100

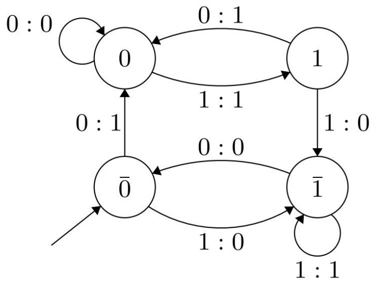


Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Let’s run the Collatz process on 321:

321	1010000010
241	1111000100



Binary representation of ancestors

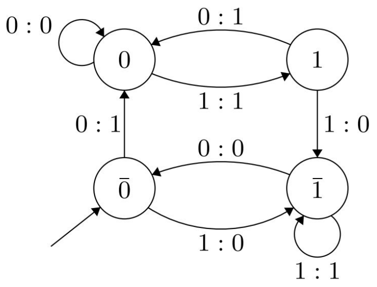
“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Let’s run the Collatz process on 642:

321 1010000010

241 1111000100

724 1011010100



Binary representation of ancestors

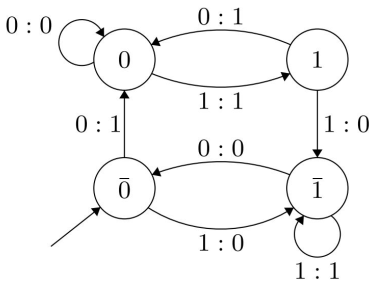
"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Let's run the Collatz process on 642:

321 1010000010

241 1111000100

181 1011010100

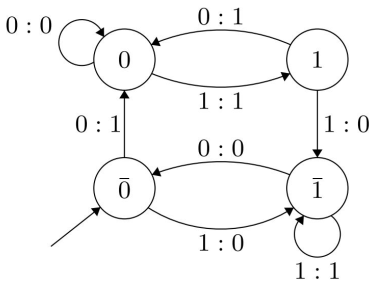


Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Let's run the Collatz process on 642:

321	1010000010
241	111100010 0
181	1011010100
544	1000100000



Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

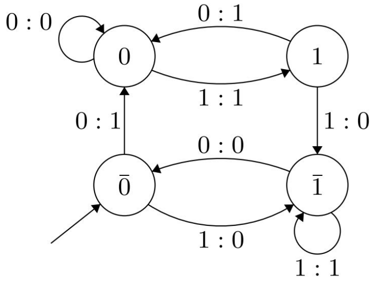
Let's run the Collatz process on 642:

321 1010000010

241 1111000100

181 1011010100

17 1000100000



642 is an ancestor of **17**

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

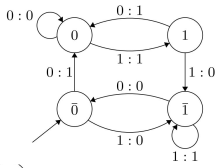
Let's run the Collatz process on 642:

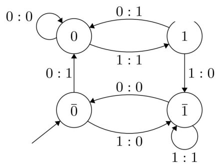
321 1010000010

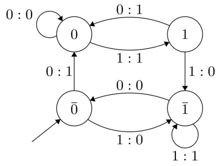
241 1111000100

181 1011010100

17 1000100000







642 is an ancestor of 17 at **"odd-distance"** k = 3

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

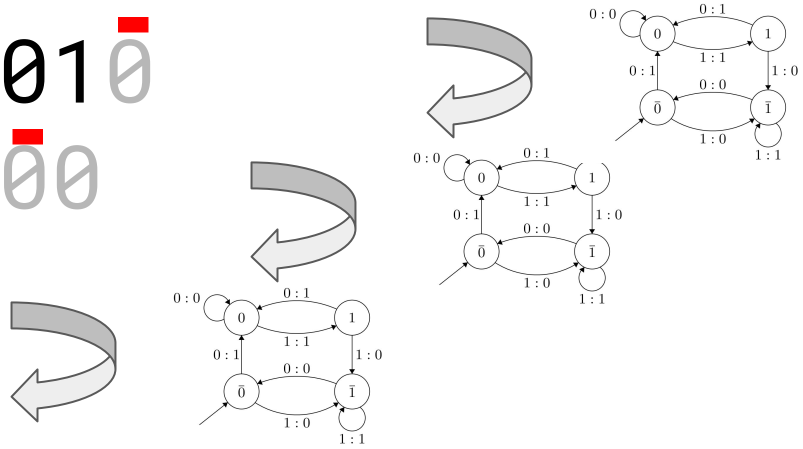
Let's run the Collatz process on 642:

321 1010000010

241 1111000100

181 1011010100

17 1000100000



Question:

What is the structure of all ancestors of 17 at "odd-distance" k = 3?

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

APPLE PATENT

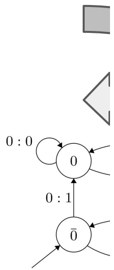
Let's run the Collatz process on 642:

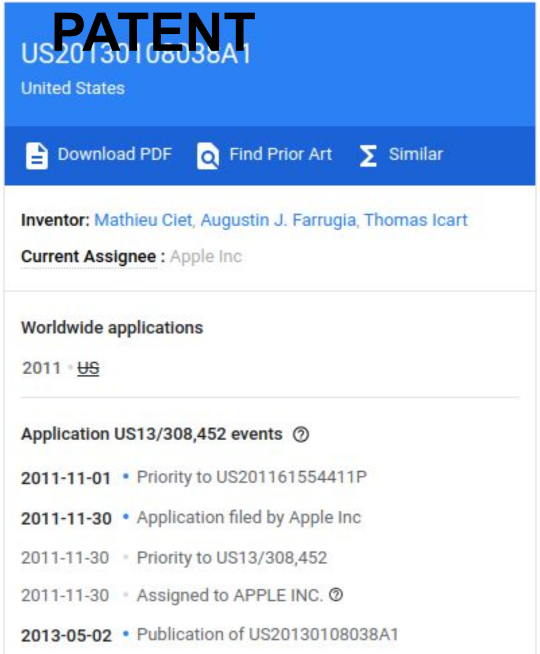
321 1010000010

241 1111000100

181 1011010100

17 1000100000





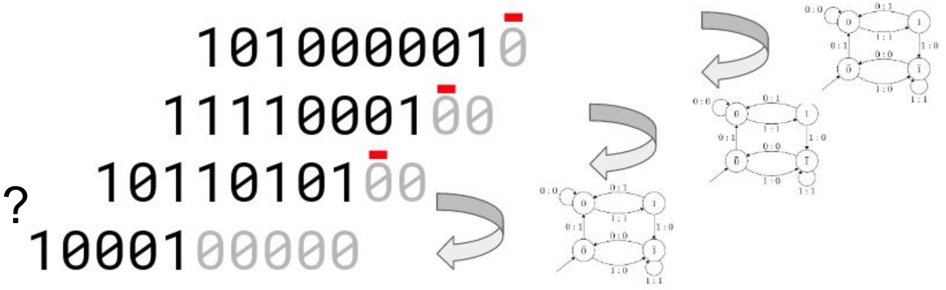
Question:

What is the structure of all ancestors of 17 at "odd-distance" k = 3?

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Question:
What is the structure of all ancestors of 17 at “odd-distance” k = 3?



What is the structure of all ancestors of x at “odd-distance” k?

Let’s denote this set by **Pred(k,x)**

THEOREM:

Pred(k,x) is regular

J. Shallit and D. A. Wilson. *Bulletin of the EACTS*, 1992.

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Question:

What is the structure of all ancestors of x at "odd-distance" k?

THEOREM:

Pred(k,x) is regular

J. Shallit and D. A. Wilson. *Bulletin of the EACTS*, 1992. ⇨ Doubly exponential regular expressions

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Question:
What is the structure of all ancestors of x at "odd-distance" k?

THEOREM:

Pred(k,x) is regular

J. Shallit and D. A. Wilson. *Bulletin of the EACTS*, 1992. → Doubly exponential regular expressions

Our Work → Simply exponential regular expressions +

Structure of these regular expressions tightly related to the structure of the Collatz process's **Parity Vectors**

Our work generalises: L. Colussi. *TCS*, 2011.
P. C. Hew. *TCS*, 2016.

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Example: ancestors of 17 at distance 3.

(1010000100101111101)*10100(000111)*00(01)*01(0)*

642 1010000010

1111000100

1011010100

1000100000

17

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Example: ancestors of 17 at distance 3.

(10100000100101111101)*10100(000111)*00(01)*01(0)*

10100000010
1111000100
1011010100
1000100000
17

10100000111000010
1111000101010100
1011010100000000
10001000000
17

Binary representation of ancestors

“Binary representation of ancestors in the Collatz graph”, T. Stérin, RP 2020

Example: ancestors of 17 at distance 3. There are **12 families.**

```
(101000010010111101)*10100(000111)*00(01)*01(0)*
(101000010010111101)*1010000(100011)*100(01)*01(0)*
(101000010010111101)*101000010010111110(110001)*1100(01)*01(0)*
(101000010010111101)*1010000100101(111000)*11100(01)*01(0)*
(101000010010111101)*10100001001(011100)*011100(01)*01(0)*
(101000010010111101)*1(001110)*0(01)*01(0)*
(101000010010111101)*10100(000111)*0001(10)*1(0)*
(101000010010111101)*1010000(100011)*10001(10)*1(0)*
(101000010010111101)*101000010010111110(110001)*110001(10)*1(0)*
(101000010010111101)*1010000100101(111000)*1(10)*1(0)*
(101000010010111101)*10100001001(011100)*01(10)*1(0)*
(101000010010111101)*1(001110)*001(10)*1(0)*
```

Code: https://github.com/tcosmo/coreli

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Result: ancestors of x at distance k. There are $2^{k-1}3^{\frac{(k-1)(k-2)}{2}}$ **families.**

▶ **Theorem 1.** *For all $x \in \mathbb{N}$, for all $k \in \mathbb{N}$ there exists a regular expression $\mathtt{reg}_k(x)$ that defines $\mathcal{E}Pred_k(x)$. The regular expression $\mathtt{reg}_k(x)$ is structured as a tree with $2^k 3^{k(k-1)/2}$ branches, alphabetic width $O(2^k 3^{k(k+1)/2})$ and star height equal to 1.*

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Result: ancestors of x at distance k. There are $2^{k-1}3^{\frac{(k-1)(k-2)}{2}}$ **families.**

▶ **Theorem 1.** *For all $x \in \mathbb{N}$, for all $k \in \mathbb{N}$ there exists a regular expression $\mathtt{reg}_k(x)$ that defines $\mathcal{E}Pred_k(x)$. The regular expression $\mathtt{reg}_k(x)$ is structured as a tree with* $\boxed{2^k 3^{k(k-1)/2}}$ *branches, alphabetic width $O(2^k 3^{k(k+1)/2})$ and star height equal to 1.*

Erratum:
k := k-1

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Example: ancestors of 17 at distance 3. There are **12 families.**

```
(101000010010111101)*10100(000111)*00(01)*01(0)*
(101000010010111101)*1010000(100011)*100(01)*01(0)*
(101000010010111101)*10100001001011110(110001)*1100(01)*01(0)*
(101000010010111101)*1010000100101(111000)*11100(01)*01(0)*
(101000010010111101)*10100001001(011100)*011100(01)*01(0)*
(101000010010111101)*1(001110)*0(01)*01(0)*
(101000010010111101)*10100(000111)*0001(10)*1(0)*
(101000010010111101)*1010000(100011)*10001(10)*1(0)*
(101000010010111101)*10100001001011110(110001)*110001(10)*1(0)*
(101000010010111101)*1010000100101(111000)*1(10)*1(0)*
(101000010010111101)*10100001001(011100)*01(10)*1(0)*
(101000010010111101)*1(001110)*001(10)*1(0)*
```

- Complexity: at least as hard as solving **discrete logarithm** in Z/3^kZ. I.e: find \`i\` such that x = 2^-i mod [3^k].

  Take: **len**()-2, here 5-2 = 3

  2^-(3) = 14^3 = 2744 = 17 mod 3^3

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Example: ancestors of 17 at distance 3. There are **12 families.**

```
(101000010010111101)*10100(000111)*00(01)*01(0)*
(101000010010111101)*1010000(100011)*100(01)*01(0)*
(101000010010111101)*10100001001011110(110001)*1100(01)*01(0)*
(101000010010111101)*1010000100101(111000)*11100(01)*01(0)*
(101000010010111101)*10100001001(011100)*011100(01)*01(0)*
(101000010010111101)*1(001110)*0(01)*01(0)*
(101000010010111101)*10100(000111)*0001(10)*1(0)*
(101000010010111101)*1010000(100011)*10001(10)*1(0)*
(101000010010111101)*10100001001011110(110001)*110001(10)*1(0)*
(101000010010111101)*1010000100101(111000)*1(10)*1(0)*
(101000010010111101)*10100001001(011100)*01(10)*1(0)*
(101000010010111101)*1(001110)*001(10)*1(0)*
```

- Complexity: at least as hard as solving **discrete logarithm** in Z/3^kZ. I.e: find \`i\` such that x = 2^-i mod [3^k].

- \`**Seeds**\` are mysterious:

  Equivalent definitions:
  - Repetend of 1/3^k in $\mathbb{Z}_2$
  - Parity bits of 1/2^k in $\mathbb{Z}/3^k\mathbb{Z}$
  - Binary expansion of $\dfrac{2^{2\cdot 3^{k-1}}}{3^k}$

Seeds approach **Full Complexity:**

J. Lòpez and P. Stoll. *Integers*, 2012

Binary representation of ancestors

"Binary representation of ancestors in the Collatz graph", T. Stérin, RP 2020

Example: ancestors of 17 at distance 3. There are **12 families.**

```
(101000010010111101)*10100(000111)*00(01)*01(0)*
(101000010010111101)*1010000(100011)*100(01)*01(0)*
(101000010010111101)*10100001001011110(110001)*1100(01)*01(0)*
(101000010010111101)*1010000100101(111000)*11100(01)*01(0)*
(101000010010111101)*10100001001(011100)*011100(01)*01(0)*
(101000010010111101)*1(001110)*0(01)*01(0)*
(101000010010111101)*10100(000111)*0001(10)*1(0)*
(101000010010111101)*1010000(100011)*10001(10)*1(0)*
(101000010010111101)*10100001001011110(110001)*110001(10)*1(0)*
(101000010010111101)*1010000100101(111000)*1(10)*1(0)*
(101000010010111101)*10100001001(011100)*01(10)*1(0)*
(101000010010111101)*1(001110)*001(10)*1(0)*
```

Futur work:

Efficiently finding the smallest ancestor at distance k?

Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Finite state automaton

Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

~~Finite state automaton~~

↓

2D Cellular automaton

Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

2D <u>Cellular automaton</u>

- 1D Base 2 and Base 3 CAs

  M. Bruschi. *preprint*, 2005

- 1D Base 6 CA

  I. Korec. *Mathematica Slovaca*, 1992

Ours: **CQCA**, 2D

Runs base 2, 3
and 6 **simulteanously**.

Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

2D <u>Cellular automaton</u>

- 1D Base 2 and Base 3 CAs
  M. Bruschi. *preprint*, 2005.
- 1D Base 6 CA
  I. Korec. *Mathematica Slovaca*, 1992.
- Quasi base 2 CA
  T. Cloney, E. Goles and G. Vichiniac. *Complex Systems*, 1987.

Eric Goles, Nicolas Ollinger,
and Guillaume Theyssier.

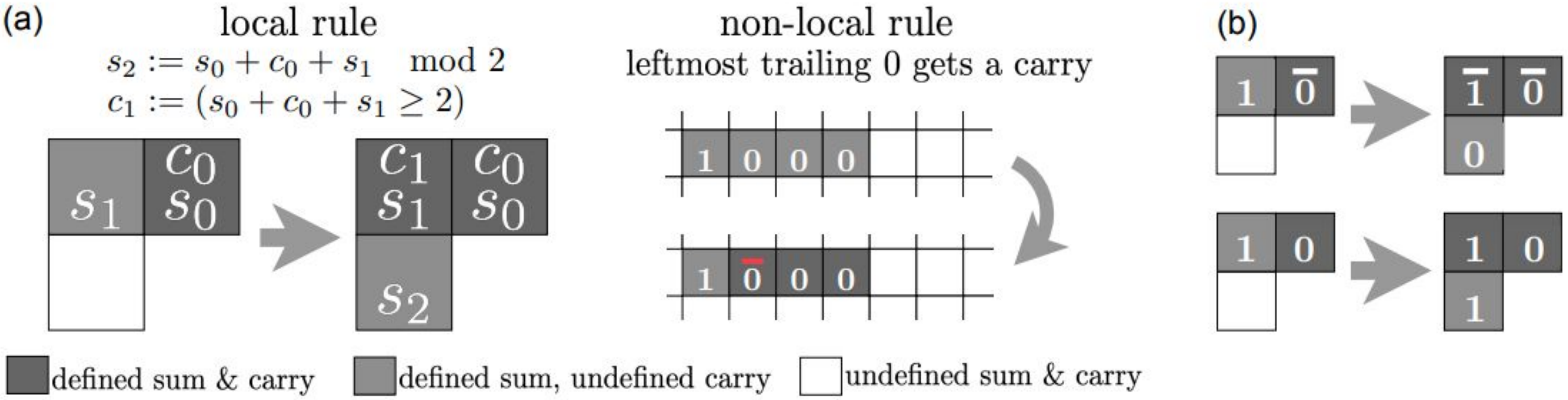
Ours: **CQCA**, 2D

Runs base 2, 3
and 6 **simulteanously**.

Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

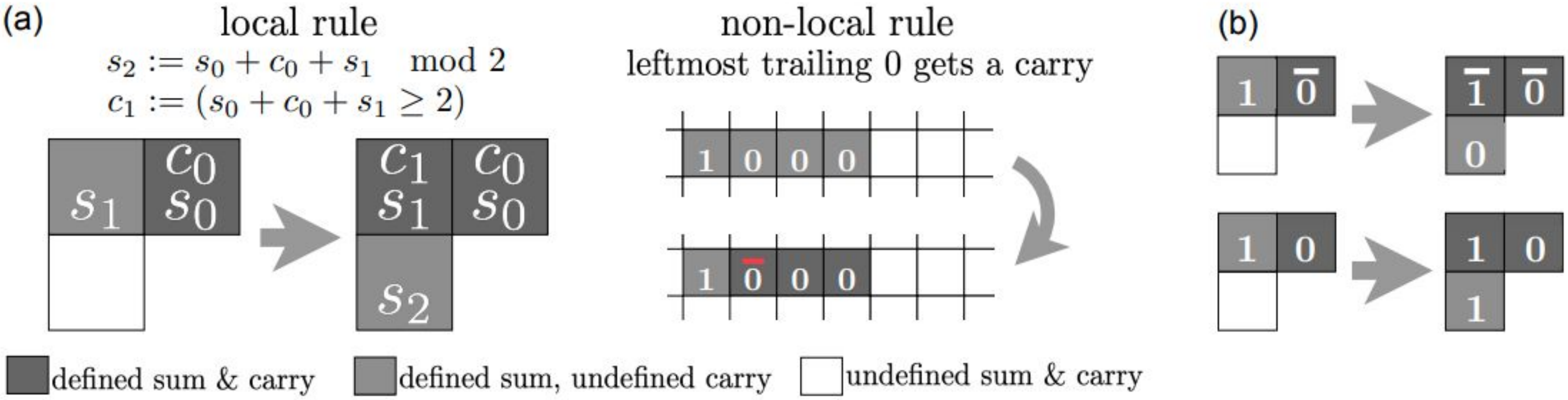
The CQCA: defined in $\mathbb{Z}^2$

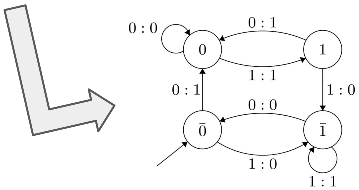


Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

The CQCA: defined in $\mathbb{Z}^2$





Simulator: `simcqca`

Code: https://github.com/tcosmo/simcqca

Base conversion: The Collatz Quasi CA (CQCA)

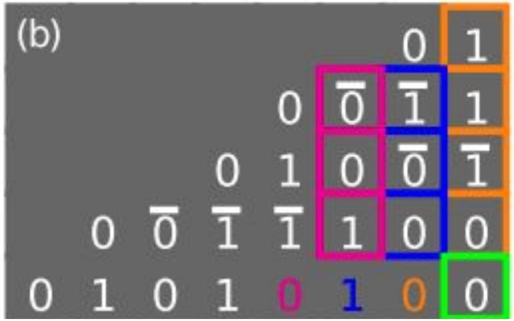
“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Base 3’:	Base 3:
0	0
$\overline{0}$	1
1	1
$\overline{1}$	2

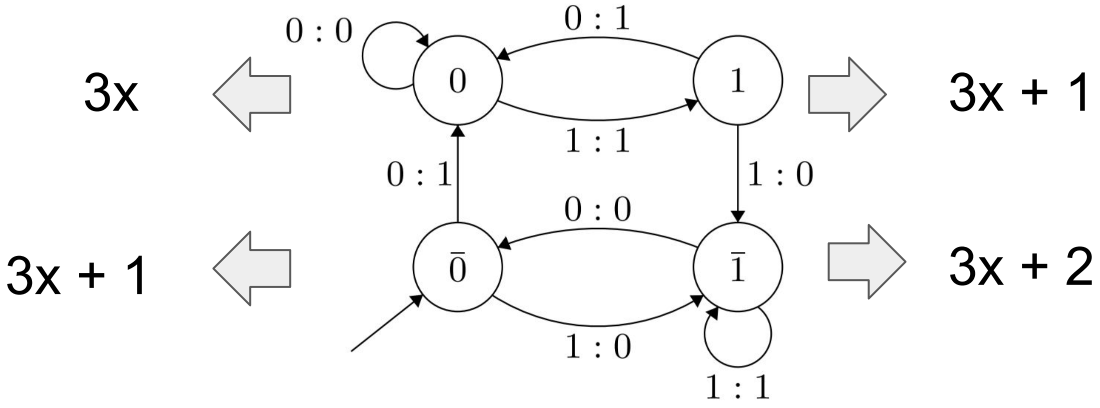
Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Base 3':	Base 3:
0	0
$\bar{0}$	1
1	1
$\bar{1}$	2



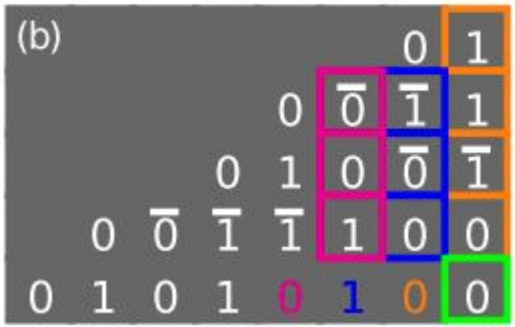
Line by Line

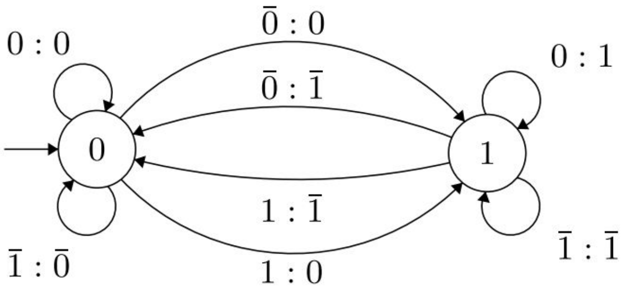


Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Column by column





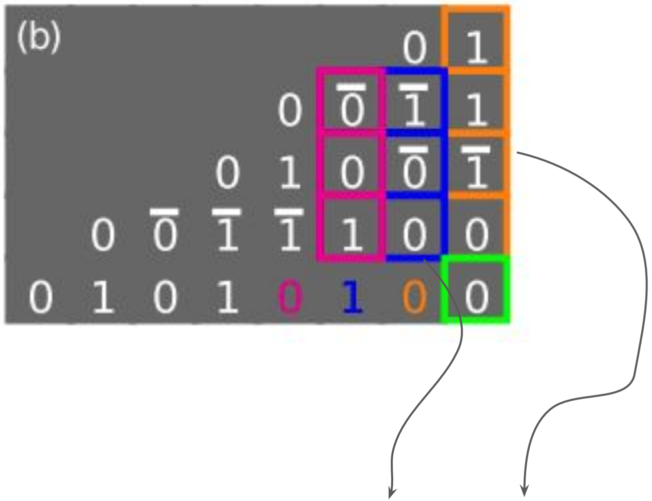
(b) The $x/2$ base $3'$ FST.

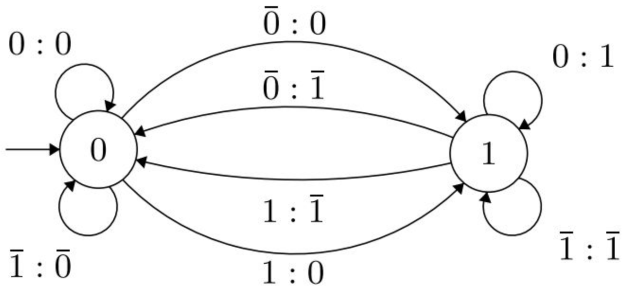
\`1120\` = 42

Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Column by column





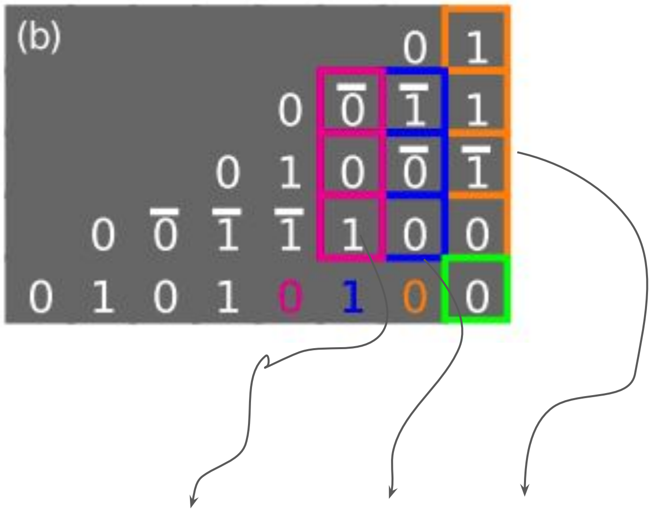
(b) The $x/2$ base $3'$ FST.

\`210\` = 21 \`1120\` = 42

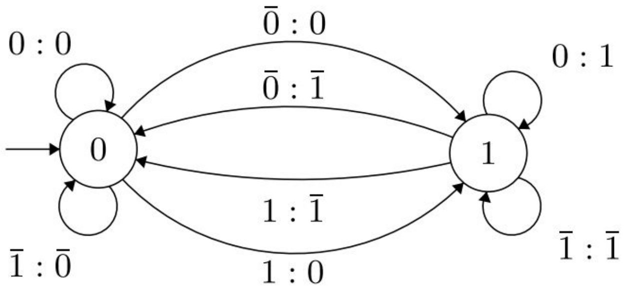
Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Column by column



`101` = 10   `210` = 21   `1120` = 42

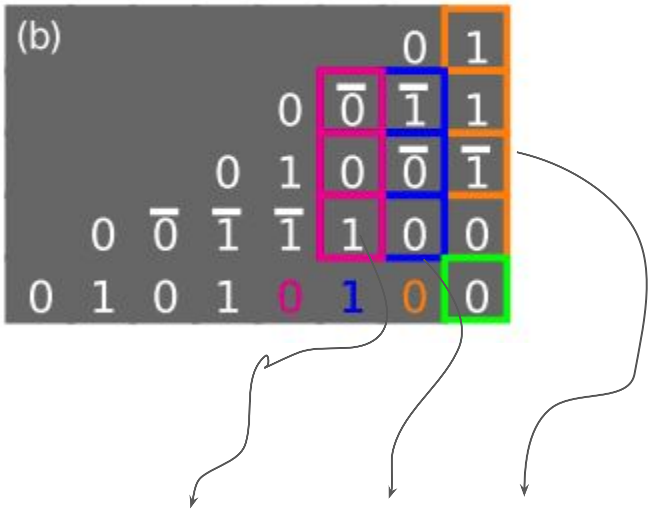


(b) The $x/2$ base $3'$ FST.

Base conversion: The Collatz Quasi CA (CQCA)

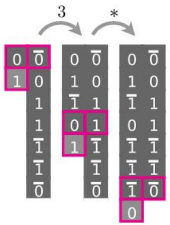
"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Column by column

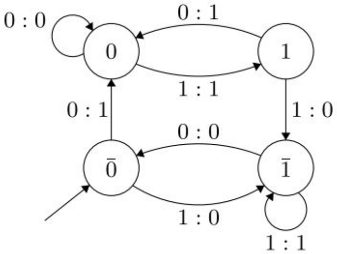


\`101\` = 10 \`210\` = 21 \`1120\` = 42

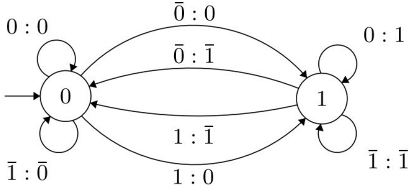
(c) Horizontal applications of the local CQCA rule simulate the $3x + 1$ FST.



(d) Vertical applications of the local rule simulate the $x/2$ FST.



(a) The $3x + 1$ binary FST.



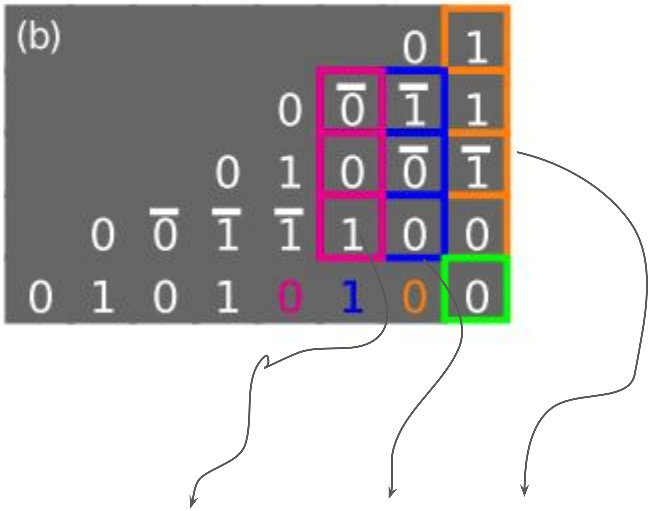
(b) The $x/2$ base $3'$ FST.

Dual Transducers

Base conversion: The Collatz Quasi CA (CQCA)

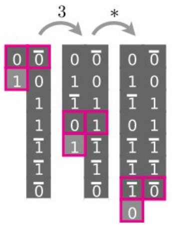
“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Column by column

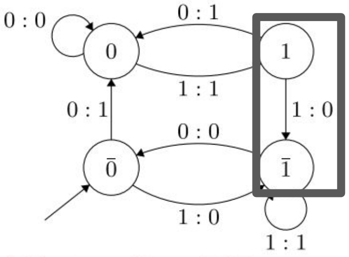


`101` = 10 `210` = 21 `1120` = 42

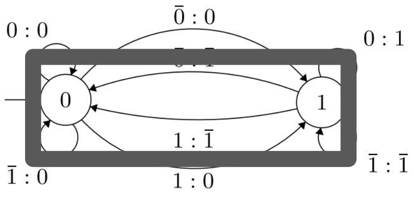
(c) Horizontal applications of the local CQCA rule simulate the $3x+1$ FST.



(d) Vertical applications of the local rule simulate the $x/2$ FST.



(a) The $3x+1$ binary FST.



(b) The $x/2$ base $3'$ FST.

Dual Transducers

Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Motivation

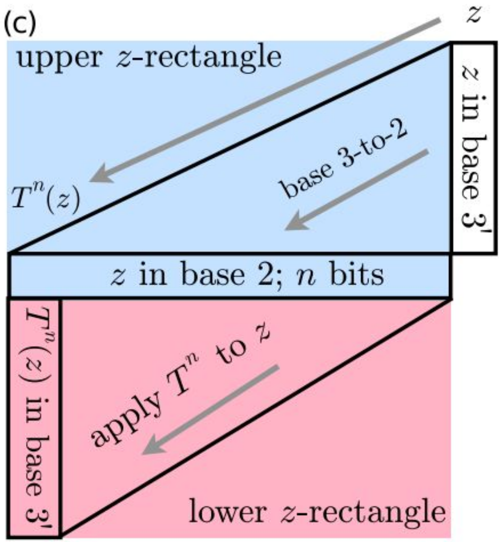
- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

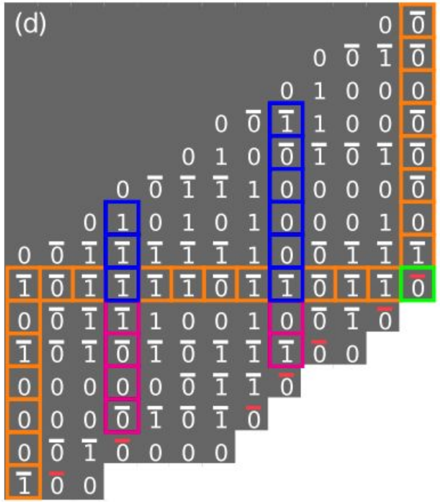
Complexity of prediction:

AC0: polysize, constant depth, arbitrary fanin, Bolean circuits

NC1: polysize, **log** depth, fanin <= 2, Bolean circuits

AC0 != P, problem PARITY not in AC0





Theorem 28 (arxiv numbering):
The upper z-rectangle prediction problem is in NC1 and not in AC0.

Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Motivation

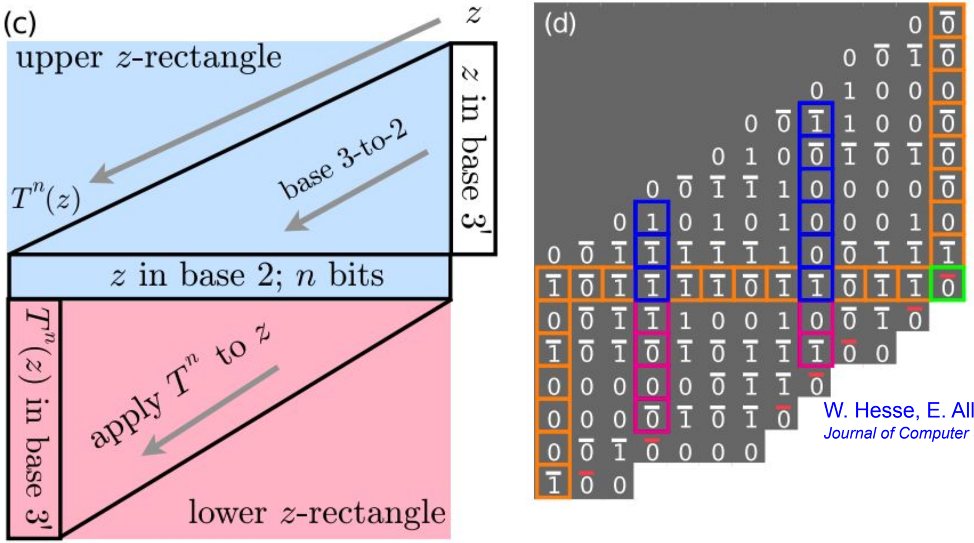
- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

Complexity of prediction:

AC0: polysize, constant depth, arbitrary fanin, Bolean circuits

NC1: polysize, **log** depth, fanin <= 2, Bolean circuits

AC0 != P, problem PARITY not in AC0



Theorem 28 (arxiv numbering):
The upper z-rectangle prediction problem is in <u>NC1</u> and not in AC0.

W. Hesse, E. Allender and D. Barrington.
Journal of Computer Science and System Science, 2002.

Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Motivation

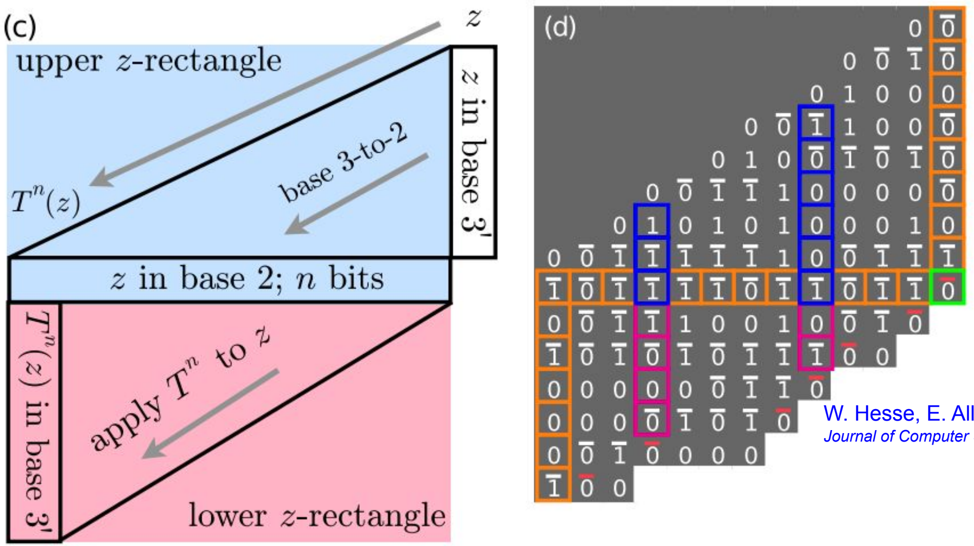
- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

Complexity of prediction:

AC0: polysize, constant depth, arbitrary fanin, Bolean circuits

NC1: polysize, **log** depth, fanin <= 2, Bolean circuits

AC0 != P, problem PARITY not in AC0



Theorem 28 (arxiv numbering):
The upper z-rectangle prediction problem is in NC1 and not in AC0.

W. Hesse, E. Allender and D. Barrington.
Journal of Computer Science and System Science, 2002.

Answering if a base 3-encoded number is odd or even reduces to PARITY.

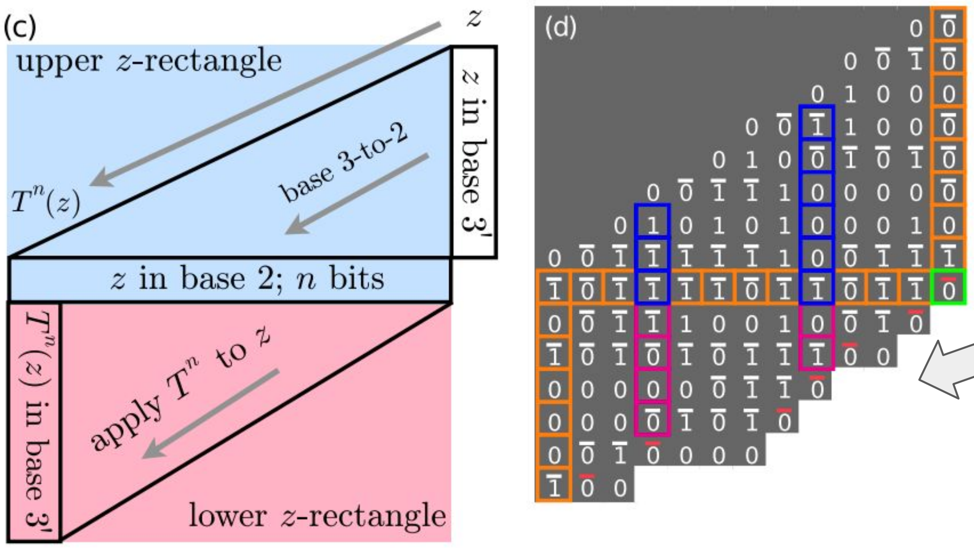
Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Motivation

- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

Future Work



Theorem 28 (arxiv numbering):
The upper z-rectangle prediction problem is in NC1 and not in AC0.

What about the lower z-rectangle?

In P, not in AC0. Can we get matching bounds?

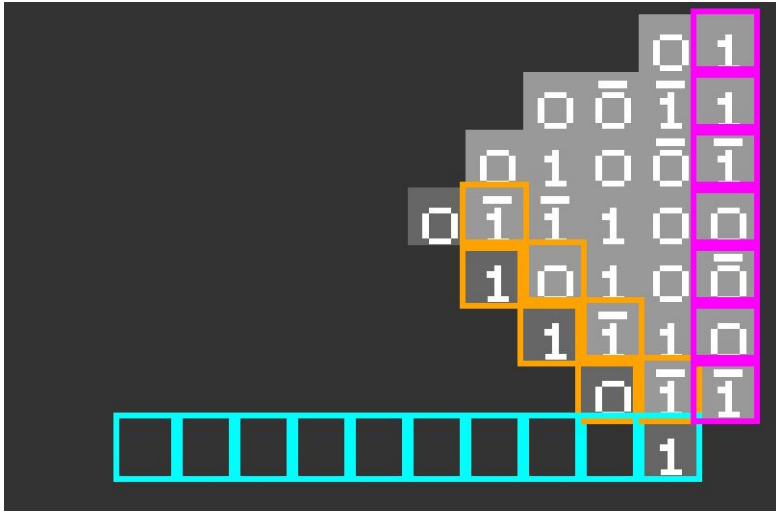
Base conversion: The Collatz Quasi CA (CQCA)

“The Collatz process embeds a base conversion algorithm”, T. Stérin and D. Woods, RP 2020

Motivation

- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

Future Work



What about this diagonal?

Magenta column: x = \`1120102\` = 1145

Blue line: x = \`10001111001\` = 1145

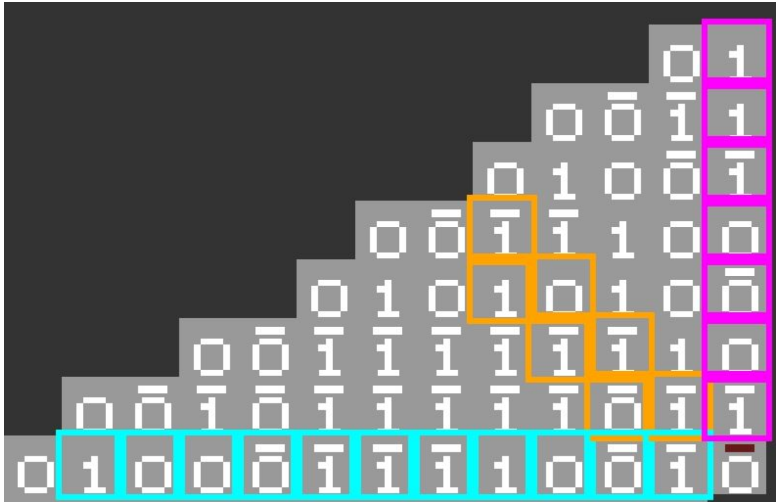
Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Motivation

- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

Future Work

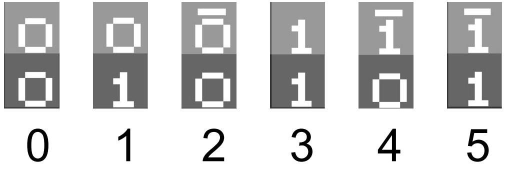


What about this diagonal?

Magenta column: x = `1120102` = 1145
Blue line: x = `10001111001` = 1145

Diagonal is x in base 6

x = `5145` = 1145



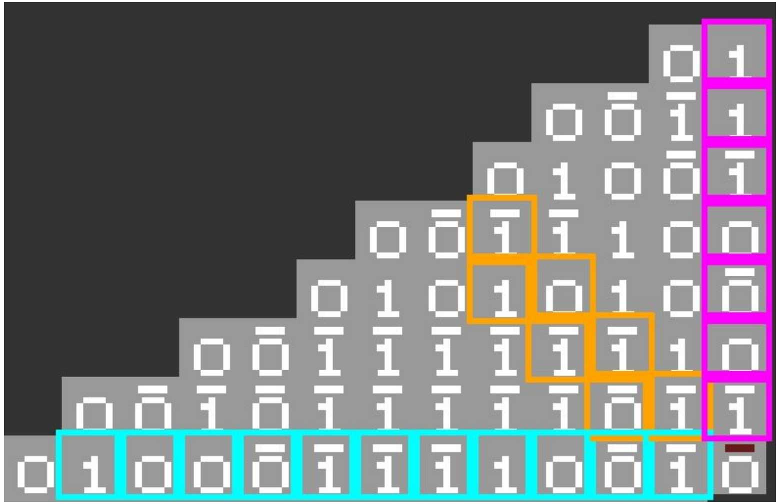
Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Motivation

- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

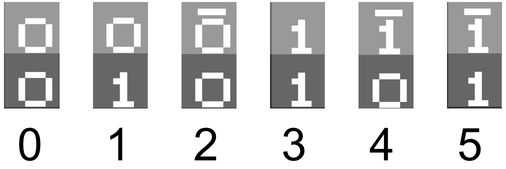
Future Work



Need to further understand this encoding

- Link with Chinese Remainder
- Link with Z2 x Z3 = Z6

x = \`5145\` = 1145



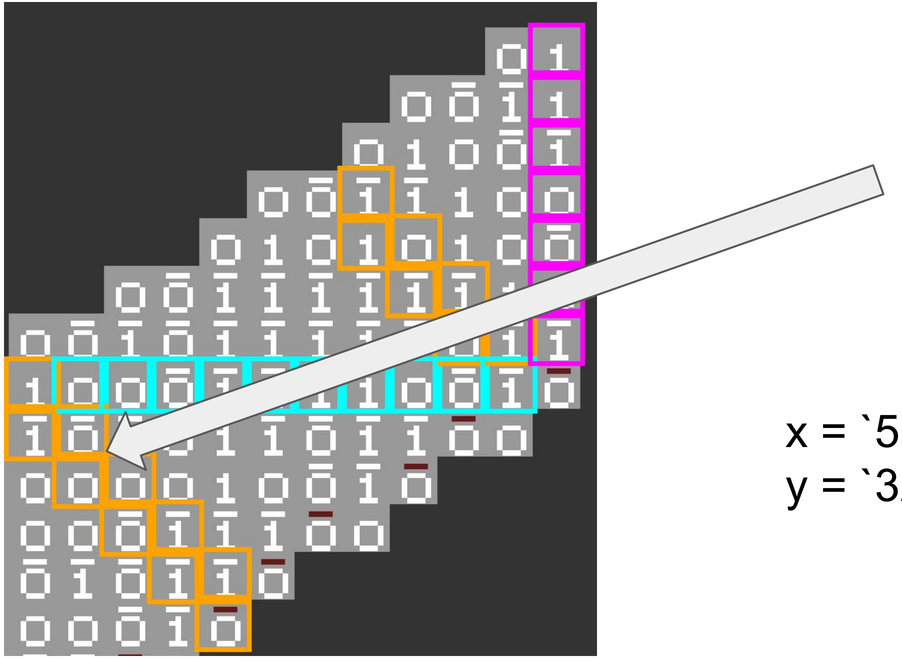
Base conversion: The Collatz Quasi CA (CQCA)

"The Collatz process embeds a base conversion algorithm", T. Stérin and D. Woods, RP 2020

Motivation

- Computational power of the Collatz process? Is it Turing complete?
- Complexity of prediction problem?

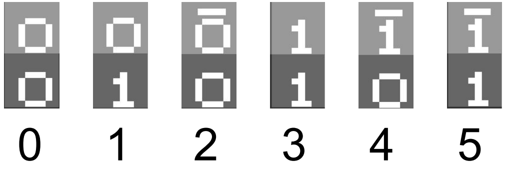
Future Work



What about that other diagonal?

- What does it compute?

x = \`5145\` = 1145
y = \`32054\` = 4354



Conclusion

- Using base 2/3/6 Finite State Automaton and Cellular Automaton descriptions of the Collatz process brings it in the scope of Computer Science.

- In that context we have a sound framework to ask:

  - What is the computational power of the Collatz process? Is it Turing complete?
  - Complexity of prediction problem?

Conclusion

- Using base 2/3/6 Finite State Automaton and Cellular Automaton descriptions of the Collatz process brings it in the scope of Computer Science.

- In that context we have a sound framework to ask:

  - What is the computational power of the Collatz process? Is it Turing complete?
  - Complexity of prediction problem?

So, what's next?

Acknowledgement

- Damien Woods, Maynooth University, Ireland
- Jose Capco, University of Innsbruck, Austria
- Jeffrey C. Lagarias, University of Michigan, USA
- Olivier Rozier, IPGP, Paris
- Anonymous reviewers

- Our institutions: Maynooth University, the Hamilton Institute & Dpt of Computer Science
- Funding agencies: European Research Council, Science Foundation Ireland

Maynooth University National University of Ireland Maynooth

Hamilton Institute

erc European Research Council Established by the European Commission

Science Foundation Ireland SFI For what's next

ERC No 772766, SFI 18/ERCS/5746