Coq proof of the 5th Busy Beaver value

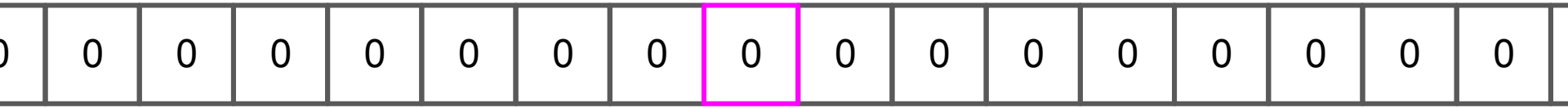
Tristan Stérin, The bbchallenge Collaboration
bbchallenge.org

ERC No 772766, SFI 18/ERCS/5746

Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

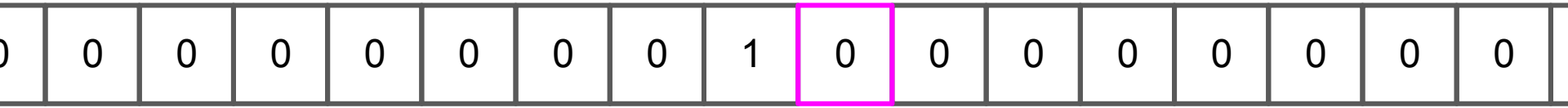
Step #0



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

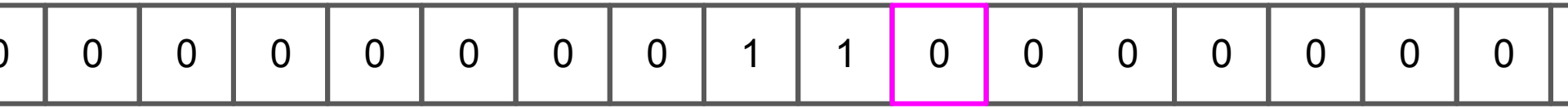
Step #1



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

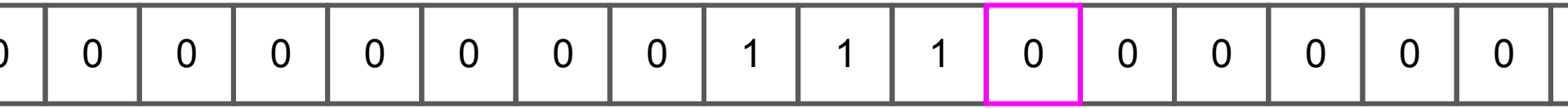
Step #2



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

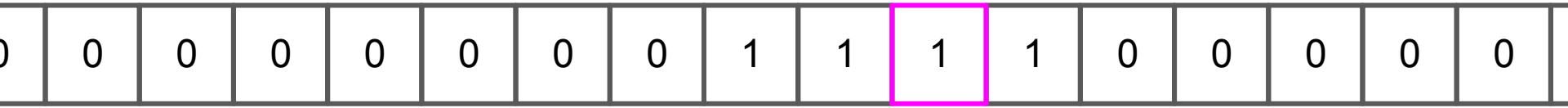
Step #3



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

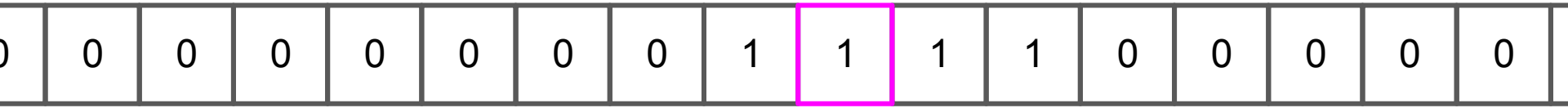
Step #4



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

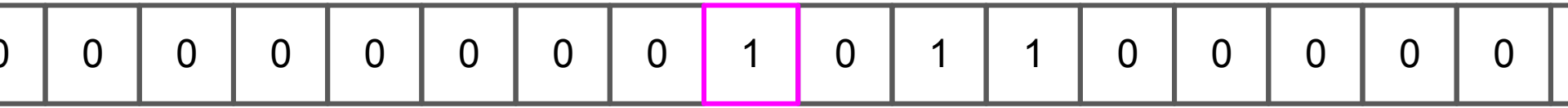
Step #5



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

Step #6



Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

Step #6

Will we ever read a 0 in state E ?

0	0	0	0	0	0	0	0	1	0	1	1	0	0	0	0	0

Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

Step #6

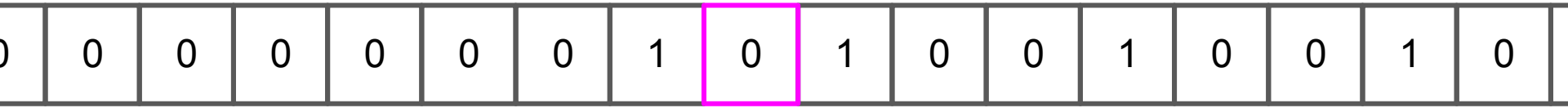
er read a 0 in state E ?

0	0	0	0	0		1	0	0	0	0	0	

Does it halt?
Yes!

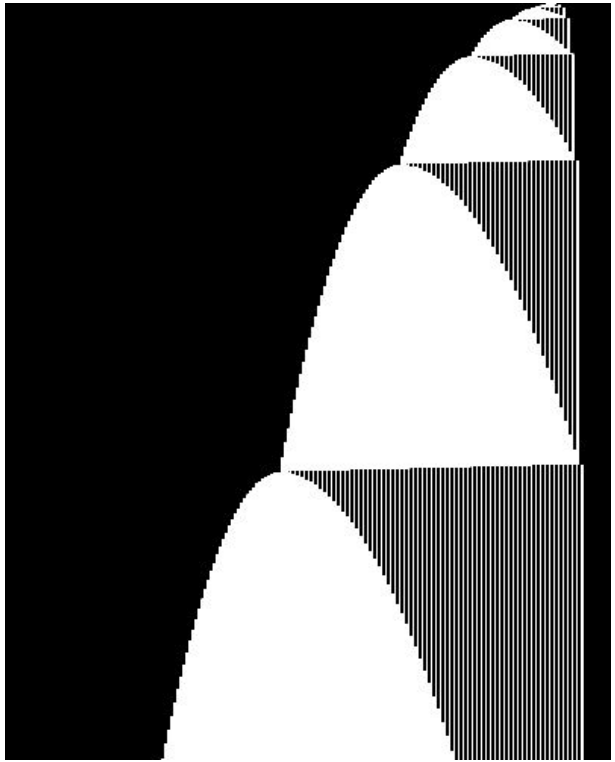
	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

Step #47,176,870



The machine has halted!!

Does it halt?
Yes!



	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

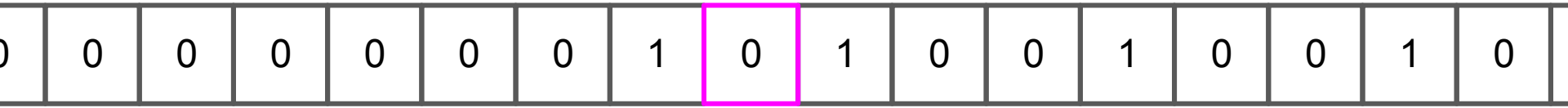
20,000-step *space-time diagram*

- Each row is a successive tape
- White = 1, Black = 0

Does it halt?

	0	1
A	1RB	1LC
B	1RC	1RB
C	1RD	0LE
D	1LA	1LD
E	---	0LA

Step #47,176,870



The machine has halted!!

Can another 5-state machine do better?

The Busy Beaver function

BB(n) = “Maximum number of steps done by a halting 2-symbol Turing machine with n states starting from all-0 memory tape”

T. Radó. On Non-computable Functions. *Bell System Technical Journal*, 41(3):877–884. 1962.

BB(n) = “Maximum algorithmic bang for your buck”

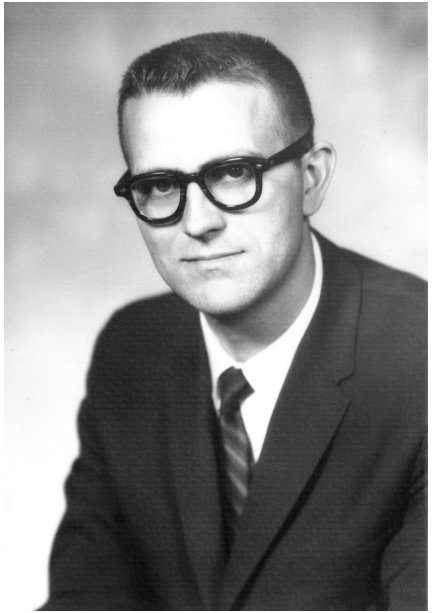
UNCOMPUTABLE

Tibor Radó, 1895 - 1965

The Busy Beaver function

BB(n) = “Maximum number of steps done by a halting 2-symbol Turing machine with n states starting from all-0 memory tape”

T. Radó. On Non-computable Functions. *Bell System Technical Journal*, 41(3):877–884. 1962.



Allen Brady, 1934 - 2024

Small busy beaver values:

- BB(1) = 1, BB(2) = 6 [Radó, 1962]
- BB(3) = 21 [Radó and Lin, 1963]
- BB(4) = 107 [Brady, 1983]

A 4-state Turing machine that runs more than 107 steps never halts (from all-0 tape)

The Busy Beaver function

BB(n) = “Maximum number of steps done by a halting 2-symbol Turing machine with n states starting from all-0 memory tape”

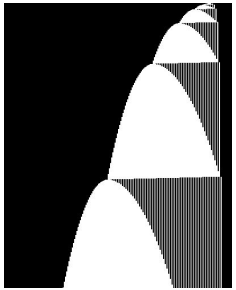
T. Radó. On Non-computable Functions. *Bell System Technical Journal*, 41(3):877–884. 1962.

Small busy beaver values:

- BB(1) = 1, BB(2) = 6 [Radó, 1962]
- BB(3) = 21 [Radó and Lin, 1963]
- BB(4) = 107 [Brady, 1983]
- BB(5) ≥ 47,176,870 [Marxen and Buntrock, 1989]

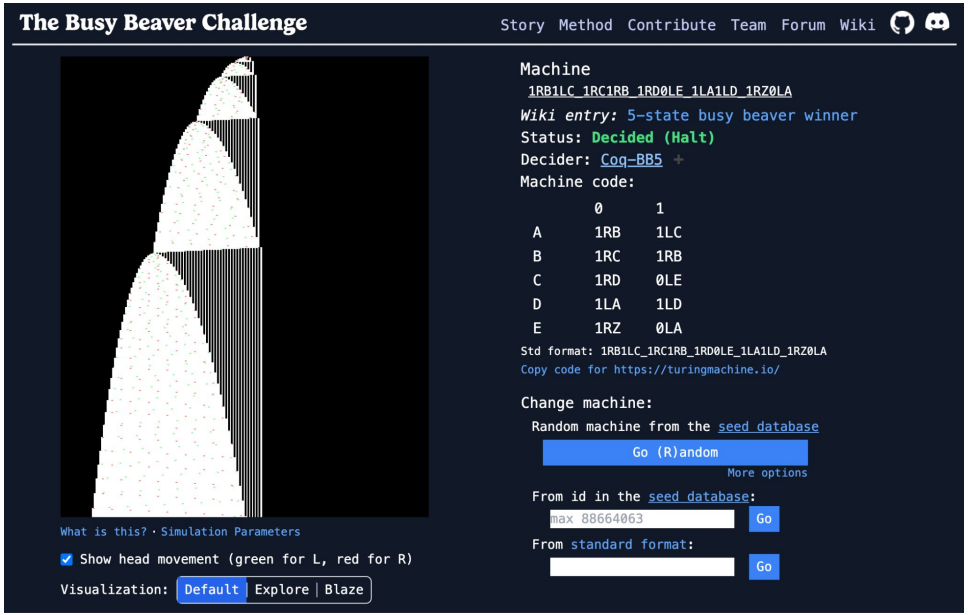
BB(5) = 47,176,870?

Conjectured yes [Aaronson, 2020]



The Busy Beaver Challenge (bbchallenge)

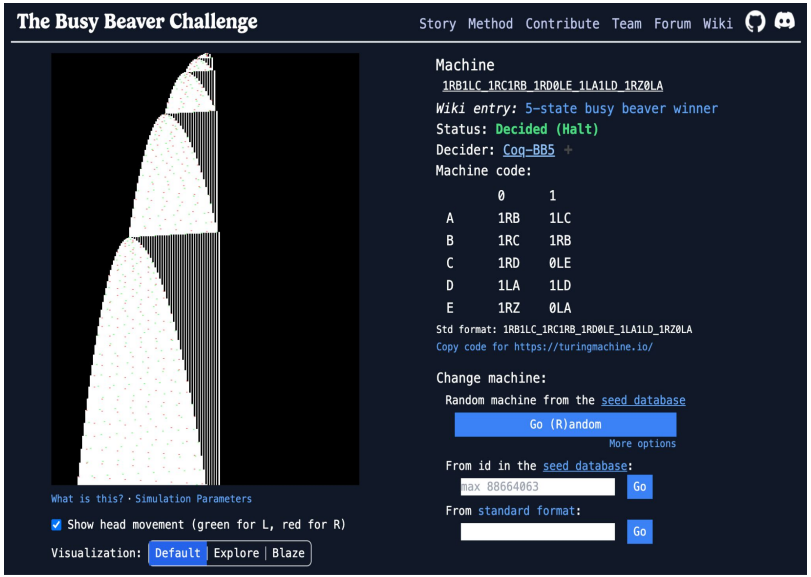
Launched in 2022



https://bbchallenge.org

- **Goal:** collaboratively solving the conjecture “BB(5) = 47,176,870”
- **Consists of:** website, Discord server, forum, wiki, *galaxy* of GitHub repositories

The Busy Beaver Challenge (bbchallenge)



https://bbchallenge.org

- Launched in 2022
- **Goal:** collaboratively solving the conjecture “BB(5) = 47,176,870”
- **Consists of:** website, Discord server, forum, wiki,
- Online, async, almost exclusively communicating on **Discord**
- No management: “**entropically-driven research**”
  - Even the formal proof happened through a grassroot initiative
- 47,000 website unique visitors, ~50 daily
- 1000+ members on Discord
- 50,000+ messages exchanged
- ~50 active contributors
- 19 contributors whose contribution made it to the Coq proof
- *Galaxy* of ~30 GitHub repositories

The Busy Beaver Challenge

How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines

The Busy Beaver Challenge

How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines
   - “Only” 181,385,789 after symmetries and pruning

The Busy Beaver Challenge

How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines
   - “Only” 181,385,789 after symmetries and pruning

2. **Craft automated proof strategies**
   - **Deciders**: programs that, given a TM output one of:

     HALTS **DOESN’T HALT** **UNKNOWN**

     → 6 deciders used in the proof

The Busy Beaver Challenge

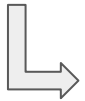
How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines
   - “Only” 181,385,789 after symmetries and pruning

2. **Craft automated proof strategies**
   - **Deciders**: programs that, given a TM output one of:

HALTS **DOESN’T HALT** **UNKNOWN**

→ 6 deciders used in the proof



Try another decider

The Busy Beaver Challenge

How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines
   - “Only” 181,385,789 after symmetries and pruning

2. **Craft automated proof strategies**
   - **Deciders**: programs that, given a TM output one of:

     HALTS **DOESN’T HALT** **UNKNOWN**

     → 6 deciders used in the proof

The Busy Beaver Challenge

How to prove “BB(5) = 47,176,870” ?

1. **Enumerate all 5-state Turing machines**
   - There are $21^{10} \approx$ 10,000 billion 5-state Turing machines
   - “Only” 181,385,789 after symmetries and pruning

2. **Craft automated proof strategies**
   - **Deciders**: programs that, given a TM output one of:

     HALTS **DOESN’T HALT** **UNKNOWN**

     → 6 deciders used in the proof

3. **Individually prove the machines that are left**

   → 13 “Sporadic Machines” in the proof

The Busy Beaver Challenge

Deciding the undecidable: this is impossible (there is no *universal* decider)

The Busy Beaver Challenge

Deciding the undecidable: this is impossible (there is no *universal* decider)

How impossible?

The Busy Beaver Challenge

Deciding the undecidable: this is impossible (there is no *universal* decider)

How impossible?

Π_1^0 -complete

There is a 25-state TM that halts iff **Goldbach's conjecture** is false [Code Golf Addict, 2016], [Leng, 2025]
There is a 744-state TM that halts iff the **Riemann hypothesis** is false [Matiyasevich and O'Rear, 2016]
There is a 748-state TM that halts iff ZF is **not consistent** [O'Rear, 2017]

The Busy Beaver Challenge

Deciding the undecidable: this is impossible (there is no *universal* decider)

Let’s do it anyway!!

(Hope was that 5-state machines do not encode unsolvable problems)

The Busy Beaver Challenge

Deciding the undecidable: this is impossible (there is no *universal* decider)

Let’s do it anyway!!

(Hope was that 5-state machines do not encode unsolvable problems)

(Or else, we would have found the simplest open problem in maths, not bad either)

The Busy Beaver Challenge

Since 2022, contributors produced dozens of deciders, in many languages:
C, C++, Go, rust, Haskell, PhP, Python

- bbchallenge is not proof assistant-centric
- At the start, I only dreamnt we'd have some formal proofs one day (I thought 5 years)

> We would encourage the use of automatic proving tools such as Lean or Coq although it would be an extremely demanding endeavour.

The Busy Beaver Challenge

Since 2022, contributors produced dozens of deciders, in many languages:
C, C++, Go, rust, Haskell, PhP, Python

- bbchallenge is not proof assistant-centric
- At the start, I only dreamnt we'd have some formal proofs one day (I thought 5 years)

> We would encourage the use of automatic proving tools such as Lean or Coq although it would be an extremely demanding endeavour.

- Instead we were using a strict validation process for deciders:
  - Requiring 2 independent implementations with matching results
  - Requiring 1 latex proof of correctness of the decider

Proof Assistants

But quickly, contributors with proof assistant skills joined:

- In 2022, Nathan Fenner started writing deciders in **Dafny**

Proof Assistants

But quickly, contributors with proof assistant skills joined:

- In 2022, Nathan Fenner started writing deciders in **Dafny**

- In 2023, Maja Kądziołka started writing deciders in **Coq**, goal was to extract them to OCaml for performance. The project is called **busycoq** and also contains the proofs of 12 out of the 13 sporadic machines

Proof Assistants

But quickly, contributors with proof assistant skills joined:

- In 2022, Nathan Fenner started writing deciders in **Dafny**
- In 2023, Maja Kądziołka started writing deciders in **Coq**, goal was to extract them to OCaml for performance. The project is called **busycoq** and also contains the proofs of 12 out of the 13 sporadic machines
- In 2024, using **Coq**, mxdys:
  - translated, improved and created new deciders
  - implemented the enumeration of 5-state TMs
  - proved the 13th sporadic machine (~10,000 lines)

That way, mxdys proved in Coq BB(5) = 47,176,870, the proof is called **Coq-BB5**

Proof Assistants

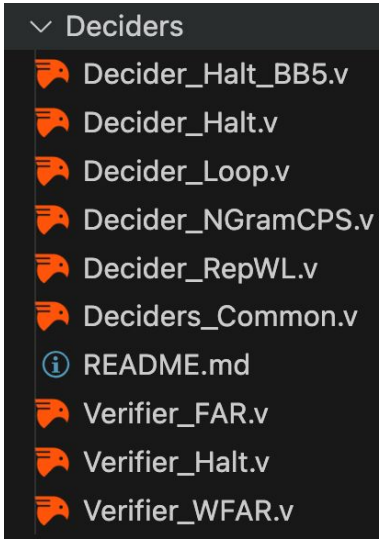
The difficult part is to prove that the deciders are correct: for each decider we need to show that if the decider says that a machine does not halt then, the machine indeed does not halt.

Also need to prove that the enumeration is correct: that the set of 181,385,789 enumerated machines is sufficient to deduce the value of BB(5).

Coq-BB5 https://github.com/ccz181078/Coq-BB5

- Released by mxdys in 2024, uses Coq v8.20
- About 20,000 lines of Coq code (+ ~10,000 lines for imported busycoq proofs)
- Implements the enumeration of TMs directly in Coq
- Contains 6 **deciders**

Decider	Method	Halt	Total
Loops	126,994,099	48,379,711	175,373,810
n-gram Closed Position Set	6,005,142		6,005,142
Repeated Word List	6,577		6,577
Halt Max (47,176,870 steps)	0	183	183
Finite Automata Reduction	23		23
Weighted Finite Automata Reduction	17		17
Sporadic Machines	13		13
1RB reduction	14		14
Total	133,005,895	48,379,894	181,385,789



Coq-BB5 https://github.com/ccz181078/Coq-BB5

- Released by mxdys in 2024, uses Coq v8.20
- About 20,000 lines of Coq code (+ ~10,000 lines for imported busycoq proofs)
- Implements the enumeration of TMs directly in Coq
- Contains 6 deciders

- Initially compiled in 13 hours (on a laptop)
- Improved to 45 minutes by Yannick Forster
  - Leverage tree structure of the enumeration to paralelise the proof
    → compilation time could be reduced arbitrarily using this technique
  - Use `coq_native` engine

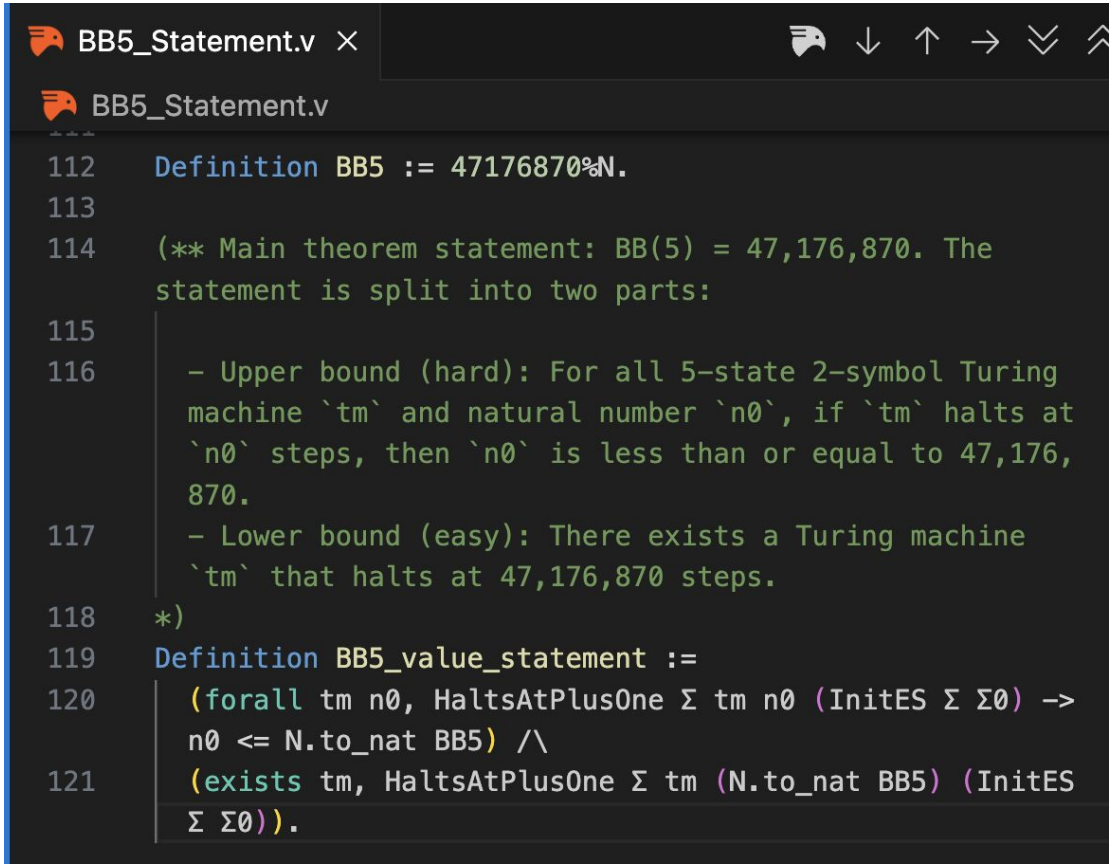
Coq-BB5 https://github.com/ccz181078/Coq-BB5

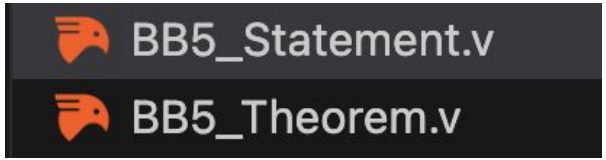
- Released by mxdys in 2024, uses Coq v8.20
- About 20,000 lines of Coq code (+ ~10,000 lines for imported busycoq proofs)
- Implements the enumeration of TMs directly in Coq
- Contains 6 deciders

- Initially compiled in 13 hours (on a laptop)
- Improved to 45 minutes by Yannick Forster
  - Leverage tree structure of the enumeration to paralelise the proof
    → compilation time could be reduced arbitrarily using this technique
  - Use `coq_native` engine

- **Output:** proof extracted to OCaml in order to output the list of enumerate machines together with halting status and decider ID.
  https://docs.bbchallenge.org/CoqBB5_release_v1.0.0/

Coq-BB5 https://github.com/ccz181078/Coq-BB5

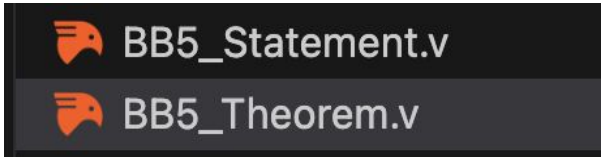




Coq-BB5 https://github.com/ccz181078/Coq-BB5

```
14  Definition BB5_champion := (makeTM BR1 CL1 CR1 BR1 DR1
    EL0 AL1 DL1 HR1 AL0).
15
16  Lemma BB5_lower_bound:
17    exists tm,
18    HaltsAt _ tm (N.to_nat BB5_minus_one) (InitES Σ Σ0).
19  Proof.
20    exists BB5_champion.
21    apply halt_time_verifier_spec.
22    vm_compute.
23    reflexivity.
24  Time Qed.
```

Lower bound: easy

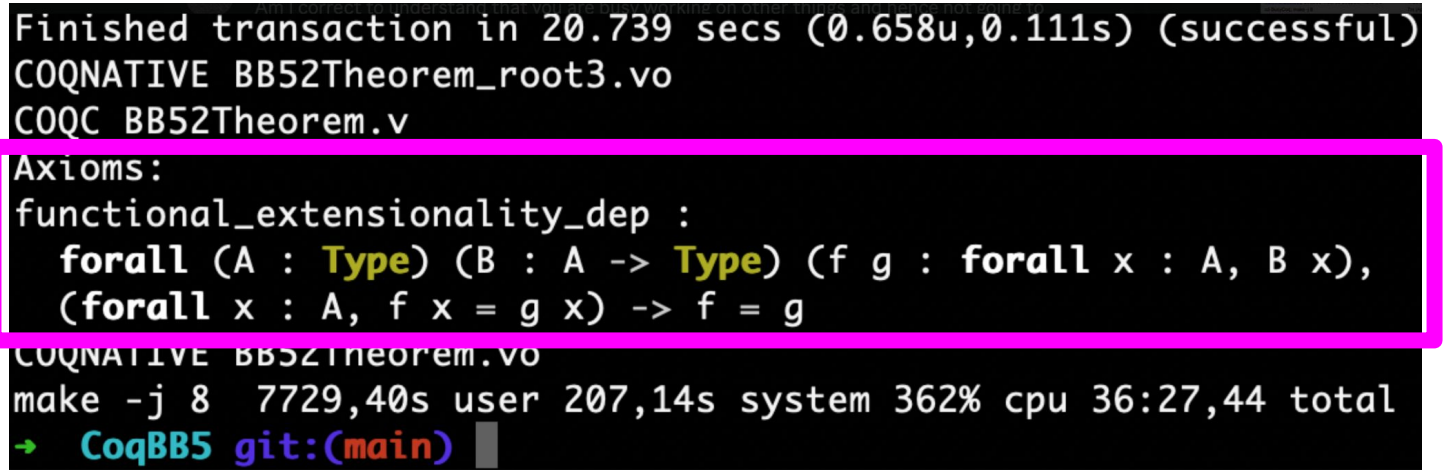



```
26  Lemma BB5_upperbound:
27    forall tm n0, HaltsAt Σ tm n0 (InitES Σ Σ0) -> n0 <= N.
      to_nat BB5_minus_one.
28  Proof.
```

Upper bound: **hard**

1. **Run enumeration**
2. **Call deciders**
3. **Solve remaining sporadic machines**

Coq-BB5 https://github.com/ccz181078/Coq-BB5



Coq-BB5 uses one standard **axiom**: functional extensionality

Summary: landscape of small BB values

Legend

Proved in Coq

Existence of a “Cryptid”

Symbols	2-State	3-State	4-State	5-State	6-State
2	$\mathrm{BB}(2) = 6$ [52]	$\mathrm{BB}(3) = 21$ [41]	$\mathrm{BB}(4) = 107$ [8]	$\mathrm{BB}(5) = 47{,}176{,}870$	$\mathrm{BB}(6) > 10 \uparrow\uparrow 15$
3	$\mathrm{BB}(2,3) = 38$ [32]	$\mathrm{BB}(3,3) > 10^{17}$	$\mathrm{BB}(4,3) > 2 \uparrow\uparrow\uparrow 2^{2^{32}}$	–	–
4	$\mathrm{BB}(2,4) = 3{,}932{,}964$	$\mathrm{BB}(3,4) > 2 \uparrow^{15} 5$	–	–	–
5	$\mathrm{BB}(2,5) > 10 \uparrow\uparrow 4$	–	–	–	–

Summary: landscape of small BB values

Legend

Proved in Coq

Existence of a “Cryptid”

Symbols	2-State	3-State	4-State	5-State	6-State
2	$BB(2) = 6$ [52]	$BB(3) = 21$ [41]	$BB(4) = 107$ [8]	$BB(5) = 47{,}176{,}870$	$BB(6) > 10 \uparrow\uparrow 15$
3	$BB(2,3) = 38$ [32]	$BB(3,3) > 10^{17}$	$BB(4,3) > 2 \uparrow\uparrow\uparrow 2^{2^{32}}$	–	–
4	$BB(2,4) = 3{,}932{,}964$	$BB(3,4) > 2 \uparrow^{15} 5$	–	–	–
5	$BB(2,5) > 10 \uparrow\uparrow 4$	–	–	–	–

Antihydra

- 6-state TM
- Simplest open problem in mathematics, on the BB scale

	0	1
A	1RB	1RA
B	0LC	1LE
C	1LD	1LC
D	1LA	0LB
E	1LF	1RE
F	---	0RA

Iterate Collatz-like function starting from 8:

- 3x/2 if even
- (3x-1)/2 if odd

Do you ever get twice as many more odd terms than even ones?

Antihydra halts iff this is true.

https://bbchallenge.org/antihydra

Future work

Legend

Proved in Coq

Existence of a “Cryptid”

Symbols	2-State	3-State	4-State	5-State	6-State
2	$BB(2) = 6$ [52]	$BB(3) = 21$ [41]	$BB(4) = 107$ [8]	$BB(5) = 47{,}176{,}870$	$BB(6) > 10 \uparrow\uparrow 15$
3	$BB(2,3) = 38$ [32]	$BB(3,3) > 10^{17}$	$BB(4,3) > 2 \uparrow\uparrow\uparrow 2^{2^{32}}$	–	–
4	$BB(2,4) = 3{,}932{,}964$	$BB(3,4) > 2 \uparrow^{15} 5$	–	–	–
5	$BB(2,5) > 10 \uparrow\uparrow 4$	–	–	–	–

Next BB values (although hopeless)

- ~4,000 machines left to solve in BB(6), over 33B total
- ~80 machines left to solve in BB(2,5)
- 6 machines left to solve in BB(3,3), including a machine that we believe **is halting**

Future work

Legend

Proved in Coq

Existence of a “Cryptid”

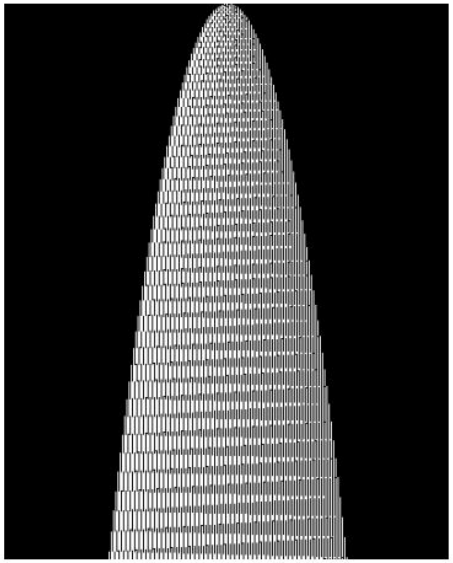
Symbols	2-State	3-State	4-State	5-State	6-State
2	$BB(2) = 6$ [52]	$BB(3) = 21$ [41]	$BB(4) = 107$ [8]	$BB(5) = 47{,}176{,}870$	$BB(6) > 10 \uparrow\uparrow 15$
3	$BB(2,3) = 38$ [32]	$BB(3,3) > 10^{17}$	$BB(4,3) > 2 \uparrow\uparrow\uparrow 2^{2^{32}}$	–	–
4	$BB(2,4) = 3{,}932{,}964$	$BB(3,4) > 2 \uparrow^{15} 5$	–	–	–
5	$BB(2,5) > 10 \uparrow\uparrow 4$	–	–	–	–

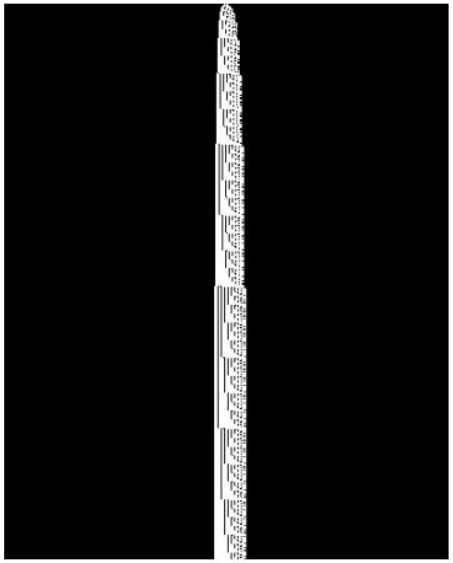
Next BB values (although hopeless)

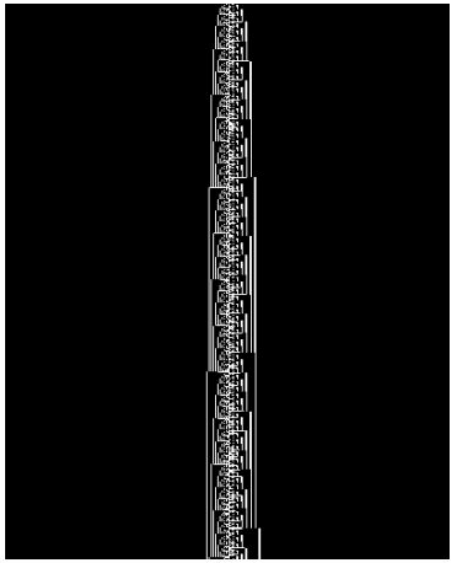
- ~4,000 machines left to solve in BB(6), over 33B total
- ~80 machines left to solve in BB(2,5)
- 6 machines left to solve in BB(3,3), including a machine that we believe **is halting**

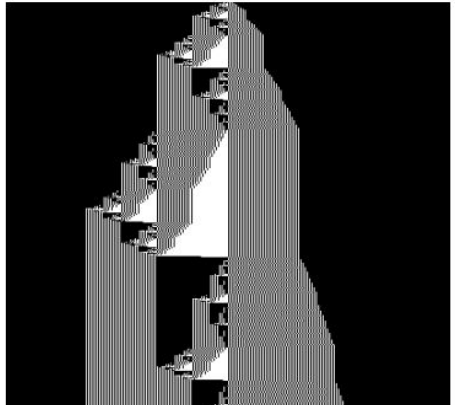
- There are still open questions about 5-state machines

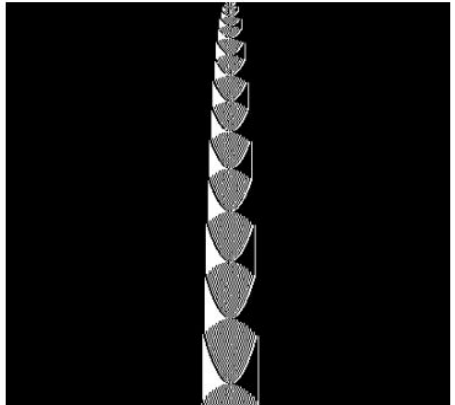
5-state machine zoo

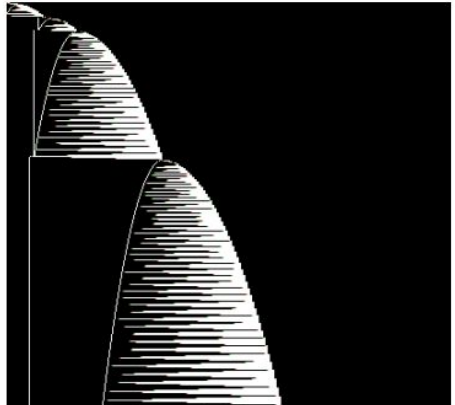




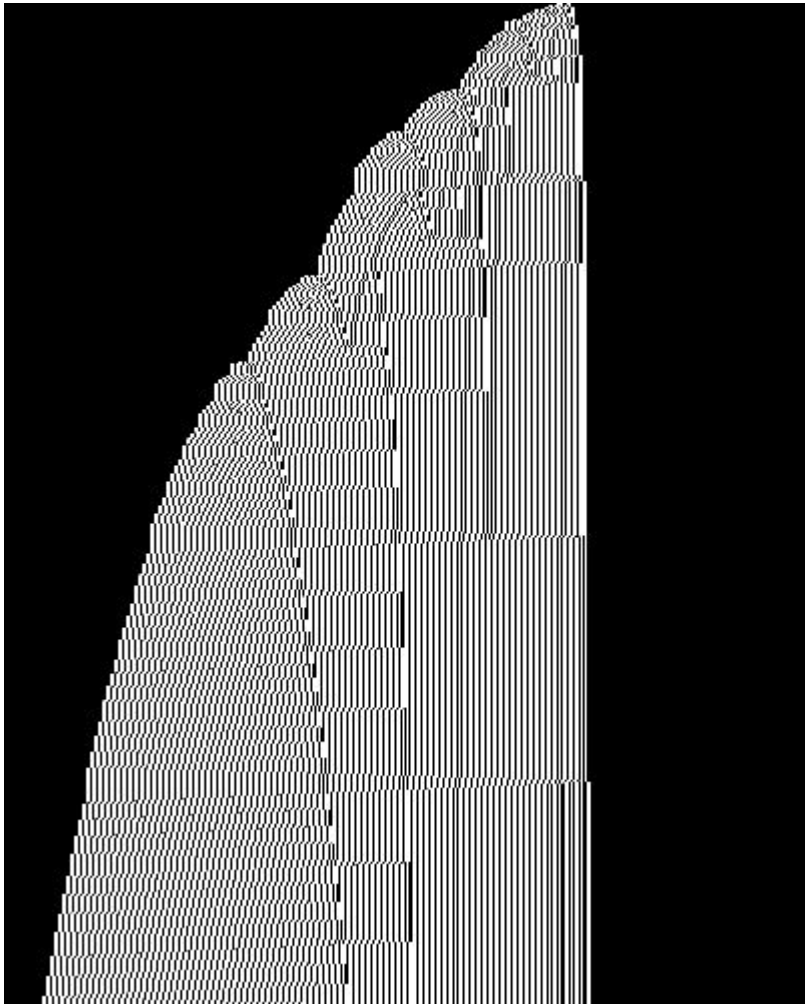








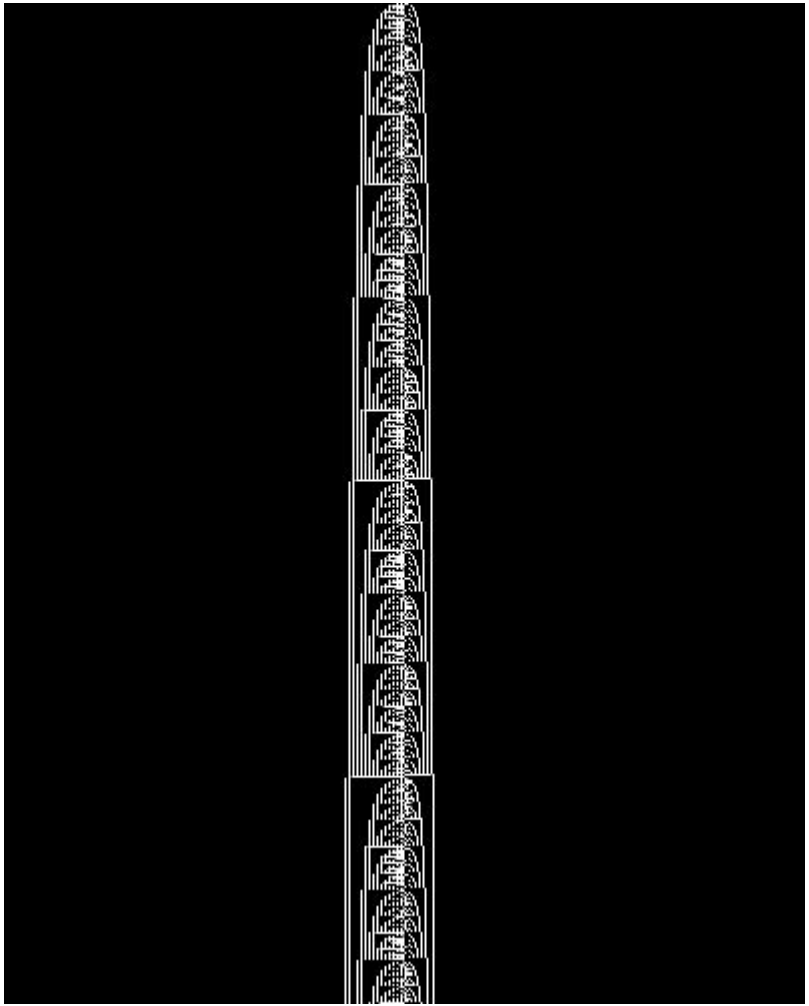
5-state
machine zoo



Machine Skelet #1

Cycles after 10^51 steps
Period is ~10^9 steps

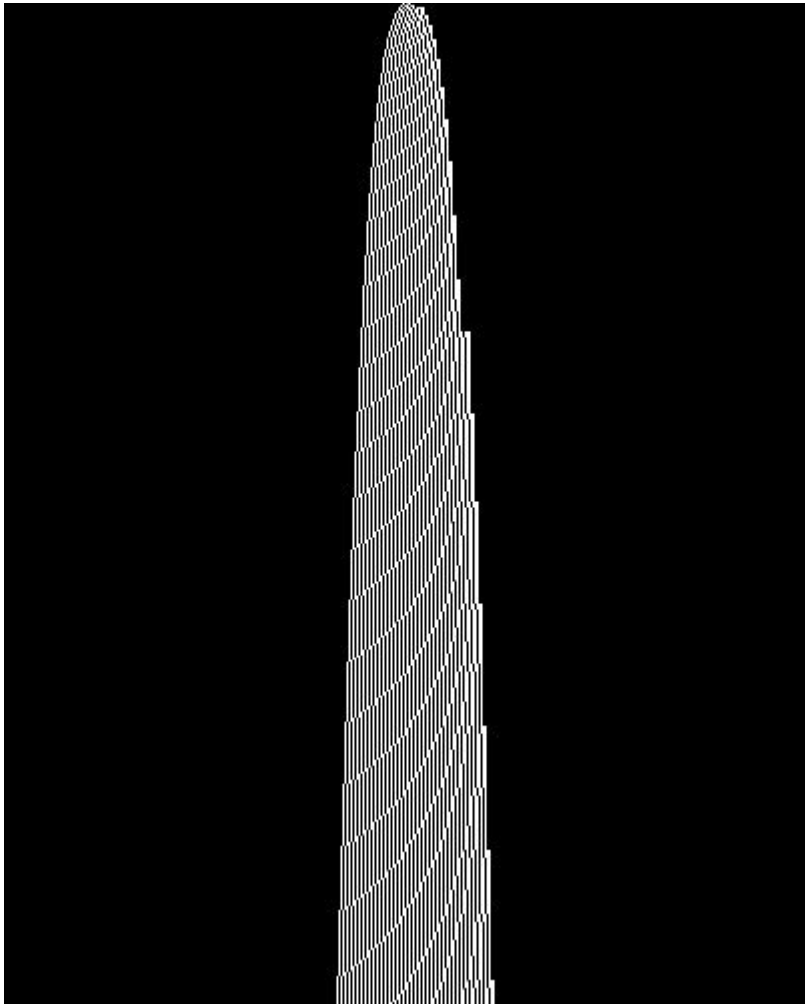
5-state
machine zoo



Machine Skelet #10

Double **Fibonacci** counter

5-state
machine zoo



Machine Skelet #17

Does Gray Code under many layers of obfuscation

Credits, many thanks to all bbchllaenge Collaborators

The bbchallenge Collaboration: $S(5)$ credits. The following contributions resulted in the determination of the fifth Busy Beaver value and in the better understanding of the landscape of small Busy Beaver values (Table 1): mxdys (Coq-BB5, Loops, RepWL); Nathan Fenner, Georgi Georgiev a.k.a Skelet, savask, mxdys (NGramCPS); Justin Blanchard, Mateusz Naściszewski, Konrad Deka (FAR); Iijil (WFAR); mei (`busycoq`); Shawn Ligocki, Jason Yuen, mei (Sporadic Machines "Shift Overflow Counters"); Shawn Ligocki, Pavel Kropitz, mei (Sporadic Machine "Skelet #1"); savask, Chris Xu, mxdys (Sporadic Machine "Skelet #17"); Shawn Ligocki, Dan Briggs, mei (Sporadic Machine "Skelet #10"); Justin Blanchard, mei (Sporadic Machines "Finned Machines"); Shawn Ligocki, Daniel Yuan, mxdys, Matthew L. House, Rachel Hunter, Jason Yuen ("Cryptids"); Yannick Forster, Théo Zimmermann (Coq review); Yannick Forster (Coq optimisation); Tristan Stérin (bbchallenge.org); Tristan Stérin, Justin Blanchard (paper writing).

Special thanks to Maja Kądziołka for some of her slides!

https://sakamoto.pl/~mei/bbslides/bbslides.html

New contributors welcome!

Thank you very much!!

Do you have any questions :) ?

The first draft of the BB5 paper is out, feedback welcome!

https://github.com/bbchallenge/bbchallenge-paper

Slides, links and more

Quanta magazine

I am open to work