Coq proof of the fifth Busy Beaver value

Tristan Stérin*[1,2] and The bbchallenge Collaboration[1]

[1] bbchallenge.org, see credits, `bbchallenge@bbchallenge.org`
[2] prgm.dev, Paris, France `tristan@prgm.dev`

Abstract

We prove that $S(5) = 47,176,870$. The Busy Beaver value $S(n)$ is the maximum number of steps a halting n-state 2-symbol Turing machine can perform from the all-0 tape before halting and S was historically introduced as one of the simplest examples of a noncomputable function. Using the Coq proof assistant, we enumerate 181,385,789 5-state Turing machines, and for each, decide whether it halts or not. Our result marks the first determination of a new Busy Beaver value in over 40 years, leveraging Coq's computing capabilities and demonstrating the effectiveness of collaborative online research.

Introduced by Tibor Radó in 1962 as part of the *Busy Beaver game* [15], $S(n)$ is the maximum number of steps that a halting n-state 2-symbol Turing machine[1] can do from the all-0 tape before halting. The function S is noncomputable as it would otherwise allow to solve the halting problem: run an n-state machine for $S(n) + 1$ steps and, if it has not halted by then, you know it will never halt.

As S in noncomputable, there is no algorithm that computes *all* of its values, but one can still try to determine *some* of them. We can get a sense of how hard this is by noting that, for instance, there is a 25-state 2-symbol Turing machine [4, 7] that iterates all even natural numbers in search of a counterexample to Goldbach's conjecture[2], and halts iff one is found (i.e. iff the conjecture is false), which means that proving the value of $S(\geq 25)$ is at least as hard as proving or disproving Goldbach's conjecture. Worse, this idea can be generalised to any other Π_1 statement[3], such as Riemann Hypothesis or ZF's consistency which have been respectively reduced to $S(744)$ and $S(748)$ [20, 19, 14, 19, 1, 6].

Nonetheless, researchers have embarked on the quest of finding what is the biggest value of S we can know. Motivations include: (i) trying to find the simplest open problem in mathematics on the Busy Beaver scale by exhibiting an n-state Turing machine for which it is unknown whether it halts or not but $S(n-1)$ is known (i.e. the halting status of all $n-1$-state Turing machines is known) and (ii) studying compute *in the wild* by discovering non-human-engineered algorithms that do *quirky things* in a *quirky way*.

Prior to this work, only the four first values of S had been proved: $S(1) = 1$, $S(2) = 6$ [15], $S(3) = 21$ [9], $S(4) = 107$ [3]. After some early attempts in the 1960s and 1970s [12, 13], the search for $S(5)$ took a turn in 1989 when Marxen and Buntrock found a new 5-state champion[4] (i.e. a 5-state machine with bigger step-count than previously known ones) achieving 47,176,870 steps [10], thus showing $S(5) \geq 47{,}176{,}870$. In 2020, after thirty years without improvements, Aaronson conjectured that there was no better 5-state machine, i.e. $S(5) = 47{,}176{,}870$ [1].

Our main result is to prove this conjecture, using the Coq proof assistant[5] [17]:

Theorem (`Lemma BB5_value`)**.** $S(5) = 47{,}176{,}870$.

The proof, called Coq-BB5, is available at `https://github.com/ccz181078/Coq-BB5`.

---

*Presenting author

[1] Turing machines with one bi-infinite tape and allowing undefined transitions.

[2] The conjecture states that every even natural number $n > 2$ can be written as the sum of two primes. This is one of the oldest open problems in mathematics.

[3] i.e. a logical statement of the form "For all x, $\phi(x)$" with ϕ using only bounded quantifiers meaning that a computer can check $\phi(x)$ in finite time for any x.

[4] `https://bbchallenge.org/1RB1LC_1RC1RB_1RD0LE_1LA1LD_1RZ0LA`

[5] The Coq proof assistant has been renamed Rocq.

$S(5)$ pipeline	Nonhalt	Halt	Total decided
1. Loops	126,994,099	48,379,711	175,373,810
2. n-gram Closed Position Set (NGramCPS)	6,005,142	0	6,005,142
3. Repeated Word List (RepWL)	6,577	0	6,577
4. Finite Automata Reduction (FAR)	23	0	23
5. Weighted Finite Automata Reduction (WFAR)	17	0	17
6. Long halters (simulation up to 47,176,870 steps)	0	183	183
7. Sporadic Machines, individual proofs	13	0	13
8. Reduction to `1RB`	14	0	14
Total	133,005,895	48,379,894	181,385,789

Table 1: Approximation of the $S(5)$ pipeline as implemented in Coq-BB5. All the 181,385,789 enumerated 5-state machines are decided by this pipeline, which solves $S(5) = 47{,}176{,}870$.

In this talk, we will present Coq-BB5 and describe, at various technical levels, how the proof works. The overall structure of the proof is as follows: the proof enumerates 5-state machines in *Tree Normal Form* (**TNF**) [2, 3, 10]. TNF essentially consists in enumerating partially-defined Turing machines starting from a machine with no transitions defined, each enumerated machine is passed through a pipeline of proof techniques mainly consisting of *deciders* which are algorithms trying to decide whether the machine halts or not [6]:

1. If the machine halts, i.e. meets an undefined transition, a new subtree of machines is visited for all the possible ways to fill the undefined transition.
2. If the machine does not halt, it is a leaf of the TNF tree.

The TNF enumeration terminates when all leafs have been reached, i.e. all the enumerated Turing machines have been decided and there are no more halting machines to expand into subtrees. The proof enumerates 181,385,789 machines[7] in about 45 minutes[8], and runs the pipeline summarised in Table 1 on each of them. All the algorithms (TNF enumeration and deciders) are programmed and proved correct in Coq.

The deciders essentially leave 13 *Sporadic Machines* undecided, for which we provide individual Coq proof of correctness [11, 18]. Among them, the monstruous *Skelet #1*[9], which is a translated loop (i.e. eventually repeats the same pattern translated in space) that only starts looping after 10^{51} steps with a period of more than 8 billion steps [8]. This machine is named after Georgi Georgiev (a.k.a Skelet) who first found it, as well as 42 other arduous ones, back in 2003 [5]. After all the enumerated machines are proved halting/nonhalting, the proof concludes $S(5) = 47{,}176{,}870$: Marxen and Buntrock's champion is the winner of the fifth Busy Beaver competition.

The talk will also discuss meta aspects of this result such as how were proof assistants used in practice and how the research was conducted within The bbchallenge Collaboration, a self-organised, international, online community of scientists (most without academic affiliation) gathered around the `bbchallenge.org` platform [16].

[6] We don't use the word "decider" in the usual sense of theoretical computer science, formally we mean an algorithm that (i) takes as input a Turing machine, (ii) is guaranteed to finish, and (iii) returns either `HALT`, `NONHALT` or `UNKNOWN`. Proving that a decider is correct means that, if the decider returns `HALT`/`NONHALT` then, indeed, the machine halts/does not halt.

[7] The list of enumerated machines, with halting status and decider information is made available at: https://docs.bbchallenge.org/CoqBB5_release_v1.0.0/.

[8] When compiled in parallel using `native_compute`. Unparallelised compilation with `vm_compute` takes about 13 hours (on a Macbook Pro M3 Max).

[9] https://bbchallenge.org/1RB1RD_1LC0RC_1RA1LD_0RE0LB_---1RC

The bbchallenge Collaboration (credits). The following contributions resulted in the determination of the fifth Busy Beaver value: 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, Rachel Hunter ("Cryptids"); Yannick Forster, Théo Zimmermann (Coq review); Yannick Forster (Coq optimisation); Tristan Stérin (bbchallenge.org); Tristan Stérin, Justin Blanchard (paper writing).

References

[1] S. Aaronson. The Busy Beaver Frontier. *SIGACT News*, 51(3):32–54, Sept. 2020. `https://www.scottaaronson.com/papers/bb.pdf`.

[2] A. H. Brady. *Solutions of restricted cases of the halting problem applied to the determination of particular values of a non-computable function.* PhD thesis, Oregon State University, 1964.

[3] A. H. Brady. The determination of the value of rado's noncomputable function $\Sigma(k)$ for four-state turing machines. *Mathematics of Computation*, 40(162):647–665, 1983.

[4] Code Golf Addict. list27.txt. `https://gist.github.com/anonymous/a64213f391339236c2fe31f8749a0df6`, 2016.

[5] G. Georgiev. Busy Beaver prover - `bbfind`. `https://skelet.ludost.net/bb/`, 2003. Accessed: 2024-11-25.

[6] Johannes Riebel. The Undecidability of BB(748). Bachelor's thesis, 2023. `https://www.ingo-blechschmidt.eu/assets/bachelor-thesis-undecidability-bb748.pdf`.

[7] Y. Leng. lengyijun/goldbach_tm: 25-state turing machine, Jan. 2025.

[8] S. Ligocki. Skelet #1 is infinite ... we think. Blog: `https://www.sligocki.com/2023/03/13/skelet-1-infinite.html`. Accessed: 2024-02-11.

[9] S. Lin. *Computer studies of Turing machine problems.* PhD thesis, Ohio State University, Graduate School, 1963.

[10] H. Marxen and J. Buntrock. Attacking the Busy Beaver 5. *Bull. EATCS*, 40:247–251, 1990.

[11] mei. busycoq. `https://github.com/meithecatte/busycoq/blob/master/verify/Skelet1.v`, 2023.

[12] P. Michel. The Busy Beaver Competitions. https://bbchallenge.org/~pascal.michel/bbc.html [Accessed 04-08-2024].

[13] P. Michel. The Busy Beaver Competition: a historical survey. Technical report, Sept. 2019.

[14] S. O'Rear. metamath - ZF. `https://github.com/sorear/metamath-turing-machines/blob/master/zf2.nql`, 2017.

[15] T. Radó. On non-computable functions. *Bell System Technical Journal*, 41(3):877–884, 1962. https://archive.org/details/bstj41-3-877/mode/2up.

[16] T. Stérin. bbchallenge.org, initial release, Mar. 2022.

[17] T. C. D. Team. The coq proof assistant, Dec. 2024.

[18] C. Xu. Skelet #17 and the fifth Busy Beaver number, 2024.

[19] A. Yedidia and S. Aaronson. A relatively small Turing machine whose behavior is independent of set theory. *Complex Systems*, 25(4):297–328, Dec. 2016.

[20] S. A. Yuri Matiyasevich, Stefan O'Rear. metamath - Riemann Hypothesis. `https://github.com/sorear/metamath-turing-machines/blob/master/riemann-matiyasevich-aaronson.nql`, 2016.