

Proof Assistant Languages

Mathematicians can write their proofs as code, which is checked by a computer.

The proof must be formulated precisely, requiring a complete understanding of the logic. This makes the program trustworthy.



20. All Primes ($\equiv 1 \pmod{4}$) Equal the Sum of Two Squares

Laurent Théry (in `coq-contribs/sum-of-two-square`):

```
Definition sum_of_two_squares :=  
  fun p => exists a , exists b , p = a * a + b * b.  
  
Theorem two_squares_exists:  
  forall p, prime p -> p = 2 /\ Zis_mod p 1 4 -> sum_of_two_squares p.
```

Mathematicians can collaborate without having to spend time understanding the proofs, since they can just trust the machine.

Busy Beaver Challenge*

$$(4n+1)^{(2n)}$$

The Busy Beaver problem involves checking 88,664,064 different cases, one by one.

Members, amateur and professional, collaborate on understanding the cases.

Each person develops a proof covering many similar cases, as many as possible.

Collaboration allows work to be split - research accelerates by sharing knowledge.

The community uses different tools, each with its purpose:

- Website (forum): scoreboard with the greatest challenge so far.
- Discord (chat): community discussions for mutual advice.
- Wiki (knowledge): technical knowledge is shared more clearly.
- Repository (code): users can combine code, proving larger parts of the problem.

Busy Beaver Challenge

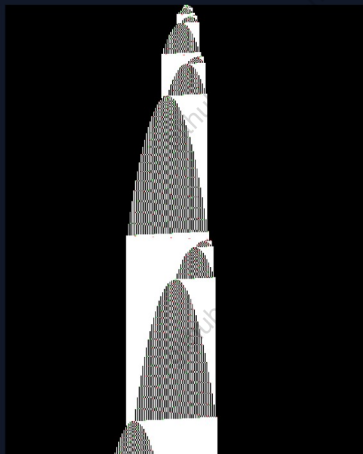
Website

Challenge goal

Setting the challenge's goal is work in progress.

Meanwhile, you can browse a [list](#) of interesting 6-state 2-symbol machines.

[Join our Discord server](#) • You can help!!



Machine

1RB0LF_1LC1LE_1RD1LB_1LC1RD_---1RF_1LA0RA

Machine code:

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

Std format: 1RB0LF_1LC1LE_1RD1LB_1LC1RD_---1RF_1LA0RA

Copy code for <https://turingmachine.io/>

[Create wiki entry](#)

Change machine:

Go (R)andom

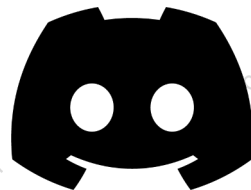
From standard format:

Go

Members receive updates on the global state of the research using a scoreboard. They can also pick patterns to study, based on look.

Community Chat

(Discord)



Busy Beaver Challenge

Wiki



[Main page](#)
[Recent changes](#)
[Random page](#)
[All pages](#)
[Help about MediaWiki](#)

[Tools](#)
[What links here](#)
[Related changes](#)
[Special pages](#)
[Printable version](#)
[Permanent link](#)
[Page information](#)

Main Page [Discussion](#)

Read [View source](#)

Main Page

The [Busy Beaver function](#) BB (called S originally) was introduced by [Tibor Radó](#) in 1962^[1] for 2-symbol [Turing machines](#) and later

BB(n,m) = Maximum number of steps taken by a halting n-state, m-symbol Turing machine starting from a blank (all 0) tape.

The 2-symbol case BB(n,2) is abbreviated as BB(n). The busy beaver function is not computable, and few of its values are known:

Small busy beaver values ^[3] ^[4]

	2-state	3-state	4-state	5-state	6-state
2-symbol	BB(2) = 6	BB(3) = 21	BB(4) = 107	BB(5) = 47,176,870	BB(6) > 10 ↑↑ 15
3-symbol	BB(2,3) = 38	BB(3,3) > 10 ¹⁷	BB(4,3) > 10 ¹⁴⁰⁷²		
4-symbol	BB(2,4) = 3,932,964	BB(3,4) > 2 ↑ ¹⁵ 5			
5-symbol	BB(2,5) > 10 ↑↑ 4				

In the above table, **cells are highlighted in orange** when there are known [Cryptids](#) (mathematically-hard machines) in that class, and Cryptid is given by using a known one with less states or symbols.

The wiki is used to share knowledge in a more precise way, building a textbook incrementally.




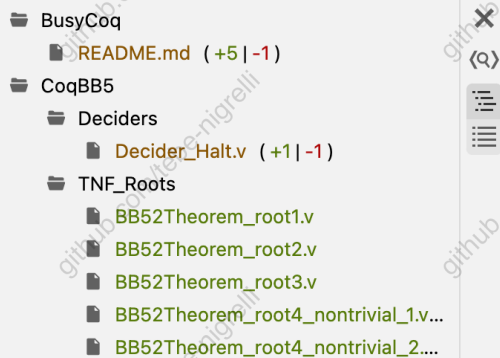
Moderators guarantee the quality of the knowledge.

New vocabulary is developed in the community:

[Probvious](#), [Hydra](#), [Bigfoot](#)

Busy Beaver Challenge

Central Code Repository

Graph	Description	Date	Author	Commit
	 origin/HEAD  origin/main Merge pull request #11 ...	24 Nov 2024...	ccz181078	a6a6eeb2
	BFS -> DFS	20 Nov 2024...	Tristan Stérin	df268be1
	NGramCPS is commented	20 Nov 2024...	Tristan Stérin	cdcb9ad8
	NGramCPS is commented	20 Nov 2024...	Tristan Stérin	9bd74fb2
	NGramCPS is commented	20 Nov 2024...	Tristan Stérin	8d99a85a
	NGramCPS is commented	20 Nov 2024...	Tristan Stérin	7b03efbe
	Moving implementation at the top and commenting it	20 Nov 2024...	Tristan Stérin	215c83ca
	pop_back and pop_back' comments	20 Nov 2024...	Tristan Stérin	fd3c59be
	comments and introducing abstraction SetOfEncodings	19 Nov 2024...	Tristan Stérin	cb5c4242
	Merge pull request #9 from yforster/main	15 Nov 2024...	ccz181078	384b4eff
	<div>Commit: 384b4eff476867c196eb00d3e6cedd6d85da...</div> <div>Parents: 05dd96e53c11b19c86113d42856b6732aec91... 0b35c9a8c525f83529b45d7f309013d44d85...</div> <div>Author: ccz181078 <1794513099@qq.com></div> <div>Committer: GitHub <noreply@github.com></div> <div>Date: Fri Nov 15 2024 18:13:14 GMT+0100 (Central European Standard Time)</div> <div>Merge pull request #9 from yforster/main</div> <div>Parallelising the BB52 proof</div> <div></div>			

Code is checked, then merged into a central repository, which slowly becomes the complete proof.

After 30 years, BB(5) was completed: now the authors are working on a human-readable proof.

The search for BB(6) and BB(2,3) continue...