Open Problems Column

William Gasarch
Univ of MD at College Park

1 This Issues Column!

This issue's Open Problem Column we revisit three prior columns for which progress was made on the open problems posed.

2 Request for Open Problems In Memory of Luca Trevisan

Luca Trevisan passed away on June 19, 2024 at the age of 52, of cancer. I am putting together an open problems column of open problems in his honor.

If you are interested in contributing then please email me a document with the following specifications.

1. It can be as short as half-a-page or as long as 2 pages. One way to make it short is to give many references or pointers to papers with more information.
2. It should be about an open problem that is either by Luca or inspired by Luca or a problem you think Luca would care about.
3. In LaTeX. Keep it simple as I will be cutting-and-pasting all of these into one column.

Deadline is Oct 1, 2024.

3 My Usual Request for Columns!

I invite any reader who has knowledge of some area to contact me and arrange to write a column about open problems in that area. That area can be (1) broad or narrow or anywhere inbetween, and (2) really important or really unimportant or anywhere inbetween.

Revisiting Three Open Problem Columns
William Gasarch

1 The Busy Beaver Function

Scott Aaronson wrote an open problems column on the Busy Beaver Function [1] in 2020. It is an excellent article that discusses the function, what is known, and why it's important. We summarize some of his article to give context for the progress made; however,

1. For a fuller account of what was known in 2020, see Scott's article referenced above.

2. For a fuller account of the breakthrough I will discuss, see Scott Aaronson's blog post on the result [2], Ben Brubaker's superb article in Quanta Magazine [6], or bbchallenge's own announcement [22]. Better yet: read all three.

Def 1.1 BB(n) is the largest number of steps that an n-state Turing machine that halts will take to halt (BB stands for Busy Beaver). Note that to define BB(n) rigorously you need to specify the details of the Turing machine model you are using. See the paper of Aaronson for these details.

It is easy to see that BB(n) is not computable. Indeed, it grows faster than any computable function.

Scott's open problems column presented the following table of known values of BB(n).

n	BB(n)	Reference
1	1	Trivial
2	6	Lin & Rado 1963 [12]
3	21	Lin & Rado 1963 [12]
4	107	Brady 1983 [5]
5	$\geq 47,176,870$	Marxen & Buntrock 1989 [13]
6	$\geq 7.4 \times 10^{36,534}$	Kropitz 2011 [11]
7	$\geq 10^{2\times 10^{x}}, x = 10^{10^{18,705,353}}$	Wythagoras 2014 (see [14, Section 4.6])

1. Kropitz in 2020 improved the lower bound on BB(6) to TOW(10, 15) which is an exponential tower of of 15 10's.

2. Kropitz's result implies that BB(7) also has that lower bound.

1.1 $\text{BB}(5) = 47,176,870$

In July of 2024 it was announced that BB(5) is indeed 47, 176, 870. We quote Scott Aaronson's blog post on the result [2]:

> Today (July 2), an international collaboration called bbchallenge [22] is announcing that it's determined, and even formally verified using the Coq [23] proof system, that BB(5) is equal to 47, 176, 870- the value that's been *conjectured* since 1990 when Heiner Marxen and Jurgen Buntrock [13] discovered a 5-state TM that runs for exactly 47, 176, 870 steps before halting, when started on a blank tape.

A personal note: normally I don't think about if the open problems column actually gets problems more out there. However in this case, it really did. I again quote Scott's Blog

> As things developed, I played no role whatsoever in the determination of BB(5) ... except for this. Tristan Sterin tells me that reading my survey article, *The Busy Beaver Frontier*, was what inspired him to start and lead the bbchallenge collaboration that finally cracked the problem. It's hard to express how gratified that makes me.

Two points on this

- Scott underrates his contribution. Writing an article that inspires research is very important.
- Since I asked Scott to write the survey, I feel I've made a contribution!

What I personally find interesting is that the number BB(5) = 47, 176, 870 isn't *that* big. I would have expected it to be enormous and possibly unknowable.

1.2 BB(6) Might Be Hopeless

There are two reasons why BB(6) may never be found.

1.2.1 Too Many Machines to Check

Lets first talk about BB(5). A while back a Busy Beaver hobbyist who goes by the name of *Skelet* claimed to have determined the status of all 5-state Turing machines except 43 of them [18]. (This result has not been refereed; however, even if the number of 5-state TM's whose status was not known was more than 43, it was not much more. We will assume that the number 43 claimed by Skelet is correct.)

Hence to find BB(5) there were only 43 machines to reason about. The difficulty is proving that a machine does not halt. The difficulty *is not* that there are lots of machines. Note that 43 is a small number.

For BB(6) there are far more machines to check. So even with the methods developed for BB(5) to prove that a particular TM does not halt (and note that they cannot always work), there are too many machines to check. Of course, there might be some unforeseen breakthrough, but I just don't see it happening.

1.2.2 Finding BB(6) Might be Equivalent to Hard Open Math Problems

Lets first talk about BB(27). A Busy Beaver hobbyist who goes by the name *Code Golf Addict* claims that there is a 27-state TM M such that

M halts iff the Goldbach Conjecture is false.

See [3] for details. (This result has not been refereed; however, there are larger values of n such that there is an n-state TM M with that property. We will assume that the machine claimed by Code Golf Addict is correct).

Hence to determine BB(27) one must solve Goldbach's conjecture. Hence BB(27) is unlikely to be known anytime soon. We note that even if Goldbach's conjecture is solved, there are other obstacles to finding BB(27).

Is there a hard open problem that is needed to solve BB(6)? For example, if solving BB(6) entailed solving the Collatz Conjecture then that would be evidence that BB(6) will not be solved anytime soon. Currently this is *not* the case. Solving BB(6) *does not* entail solving the Collatz conjecture.

However there is an open problem that is needed to solve BB(6) that is *related* to the Collatz Conjecture and *seems* hard.

Def 1.2

1. The *Collatz Function* is defined as follows:

$$f(x) = \begin{cases} \frac{x}{2} & \text{If } x \text{ is even} \\ 3x+1 & \text{If } x \text{ is odd} \end{cases} \quad (1)$$

2. The *Collatz Conjecture* is that if $n \geq 1$ then $f(n)$, $f(f(n))$, $\cdots$ is eventually 1. There is much empirical evidence for this conjecture; however, it seems hard to prove. Paul Erdős has said *Math is not ready for this problem.*

Consider the following problem:

$$8, f(8), f(f(8)), \cdots$$

Stop when the number of odd terms is bigger than twice the number of even terms. Scott Aaronson reports that Tristan Sterin [19] claims that there is a 6-state machine M such that

M **halts iff the sequence above that begins** $f(8)$ **halts.**

Hence to determine BB(6) one must solve determine if that sequence halts. This seems like a hard problem. And again, even if one solved that one problem there are many more programs to look at which may also involve hard math problems.

Open Problem 1.3 Find an $n \leq 26$ and an n-state TM M such that

M halts iff the Goldbach Conjecture is false.

If $n = 6$ that will show that BB(6) is hard to find. Of course, we already think its hard to find.

2 The BEE Sequence

Bill Gasarch & Emily Kaplitz & Erik Metz [9] studied the following sequence.

Def 2.1 The BEE sequence is as follows:
$a_1 = 1$
$(\forall n \geq 2)[a_n = a_{n-1} + a_{\lfloor n/2 \rfloor}]$.
(BEE stands for Bill-Emily-Erik.)

They proved the following (some of which was already known).

Theorem 2.2

1. *Let* $m \in \{2, 3, 5, 7\}$. $(\exists^\infty n)[a_n \equiv 0 \pmod{m}]$.
2. *Let* $m \equiv 0 \pmod{4}$. $(\forall n)[a_n \not\equiv 0 \pmod{m}]$.

Based on a lot of empirical evidence they made the following conjecture.

Conjecture 2.3 *Let* $m \in \mathbb{N}$ *such that* $m \not\equiv 0 \pmod{4}$. *Then* $(\exists^\infty n)[a_n \equiv 0 \pmod{m}]$.

Van Doorn [21] generalized and expanded the questions raised by Gasarch & Kaplitz & Metz.

Def 2.4 Let $0 \leq x \leq m - 1$.

1. $S_{x,m}(n) = |\{k : (1 \leq k \leq n) \wedge (a_k \equiv x \pmod{m})\}|$.
2. $d_{x,m} = \lim_{n\to\infty} \frac{S_{x,m}(n)}{n}$. ($d_{x,m}$ might not exist.)
3. $\underline{d_{x,m}} = \liminf_{n\to\infty} \frac{S_{x,m}(n)}{n}$. ($\underline{d_{x,m}}$ always exists.)

By Theorem 2.2 the following is known.

1. Let $r \in \{2, 3, 5, 7\}$. Then $\lim_{n\to\infty} S_{0,r}(n) = \infty$.
2. Let $r \equiv 0 \pmod{4}$. Then $(\forall n)[S_{0,r}(n) = 0]$.

Van Doorn proved the following.

Theorem 2.5

1. *If* $x \in \{1, 2, 3, 5, 6, 7\}$ *then* $S_{x,8}(n) \geq \frac{n}{6} - 2\log(n) - 11$.
2. *If* $x \in \{1, 2, 3, 5, 6, 7\}$ *then* $d_{x,8} = \frac{1}{6}$. *(This follows from Part 1.)*
3. *There is an algorithm that does the following:*

- **Input:** $(x, m) \in \mathbb{N} \times \mathbb{N}$ *such that* x *and* m *are not both divisible by 4, and such that* x *is divisible by the largest odd divisor of* m.
- **Output:** *A lower bound on* $\underline{d_{x,m}}$. *Note that the lower bound might be 0 which is not interesting. However, as you will see in the next point, the algorithm has been used to obtain some nontrivial lower bounds.*

4. *Using the algorithm in the last part, the following lower bounds were obtained:*

    (a) $\underline{d_{0,3}} > 0.2873$
    (b) $\underline{d_{0,5}} > 0.1148$
    (c) $\underline{d_{0,6}} > 0.0689$
    (d) $\underline{d_{3,6}} > 0.1773$
    (e) $\underline{d_{0,7}} > 0.0890$
    (f) $\underline{d_{0,9}} > 0.0509$
    (g) $\underline{d_{0,10}} > 0.0280$
    (h) $\underline{d_{5,10}} > 0.0862$
    (i) $\underline{d_{0,11}} > 0.0392$
    (j) $\underline{d_{3,12}} > 0.0783$
    (k) $\underline{d_{6,12}} > 0.0689$
    (l) $\underline{d_{9,12}} > 0.0783$
    (m) $\underline{d_{0,13}} > 0.0118$
    (n) $\underline{d_{0,14}} > 0.0107$
    (o) $\underline{d_{7,14}} > 0.0430$
    (p) $\underline{d_{0,15}} > 0.0039$

What I personally find interesting about these numbers is that they are reasonable. When I hear a statement like $d_{0,13}$ *is nonzero* I expect something like $d_{0,13} > \frac{1}{10^{40}}$. Hence I am delighted that the worst bound is 0.0039.

Van Doorn has emailed me the following:

And in my paper I mentioned that I have actually checked with a computer for all $m \leq 15$. *But please encourage your readers to improve upon this! Simply write a better computer program than what I came up with in order to check more cases and get better lower bounds.*

Van Doorn has some speculation and tenuous conjecture. If you are interested then read his paper.

3 A Sequence from an Oliver Roeder Column

Gasarch wrote an an Open Problems Column [10] about a sequence from Oliver Roeder's Riddler Column.

Def 3.1 Let $n \in \mathbb{N}$. A sequence is n-linked if

1. Every element in the sequence is in $\{1, \ldots, n\}$.
2. No element appears more than once in the sequence.
3. Every element is either a factor or multiple of the previous element (except the first element which has no previous element).

Example: The following is a 100-linked sequence of length 17.
17, 1, 47, 94, 2, 4, 20, 5, 10, 90, 3, 21, 7, 42, 6, 30, 15

Open Problem 3.2 Let $k(n)$ be the length of the longest linked n-sequence. Get upper and lower bounds on $k(n)$ asymptotically.

The following problem was implicit.

Open Problem 3.3 It is known that $k(100) = 77$ by a computer program.

1. Find a human-readable proof of the result $k(100) = 77$.
2. For (say) $101 \leq k \leq 200$ find $k(n)$. While human-readable proofs are preferred, computer proofs are fine.

We note that the term *human-readable proof* is not well defined. Indeed, if a human has a 10-page proof of boring casework then I do not know if I count that as human-readable.

After the paper appeared I began working on the open problems with David Harris and his son Tomas. We later found that a lot was known about it. See the blog posts [8] (or just read the rest of this section) for what was known, and [7] for how we did not know these results when I wrote the column (Spoiler Alert-Some of the papers were in French but some of the fault is mine.)

Here is what was already known:

1. In 1983 Pollington [15] proved $k(n) \geq ne^{-\mathrm{polylog}(n)}$.
2. In 1983 Pomerance [16] proved $K(n) \leq o(n)$.
3. In 1995 Tenenbaum [20] proved (in a paper written in French) that there exists a, b such that

$$\frac{n}{(\log n)^a} \leq k(n) \leq \frac{n}{(\log n)^b}$$

4. In 2021 Saias [17] showed, in a paper written in French, that

$$k(n) \geq (0.3 - o(1))\frac{n}{\log n}.$$

These results use hard and interesting math. As such they take the paper out of the realm of the Riddle Column, which is for recreational math, and into the realm of serious math.

Gaetan Berthe emailed me a description of a program that he wrote with Paul Revenant which finds actual values for $k(n)$. For that description, see the blog post [8]. He also sent me a table [4] of the first 1000 values of $k(n)$. Gaetan pointed out that these numbers have not been refereed; however, they are surely correct.

References

[1] S. Aaronson. The busy beaver frontier (an open problems column). *SIGACT News*, 53(3):31–55, 2020.
`https://www.cs.umd.edu/users/gasarch/open/busybeaver.pdf`.

[2] S. Aaronson. BusyBeaver(5) is now known to be 47,176,870, 2024.
`https://scottaaronson.blog/?p=8088`.

[3] C. G. Addict. No name, 2016.
`https://gist.github.com/anonymous/a64213f391339236c2fe31f8749a0df6` is a pointer to a github site with code.

[4] G. Berthe. No title, 2020.
`{https://www.cs.umd.edu/~gasarch/BLOGPAPERS/1-1000.txt}` Year is uncertain.

[5] A. H. Brady. The determination of the value of Rado's noncomputable function $\Sigma(k)$ for four-state Turing machines. *Mathematics of Computation*, pages 647–665, 1983.

[6] B. Brubaker. With fifth Busy Beaver, researchers approach computation's limits. *Quanta*, 2024.
`https://www.quantamagazine.org/amateur-mathematicians-find-fifth-busy-beaver-turing-`

[7] W. Gasarch, D. Harris, and T. Harris. The Roeder seq problem was solved before I posted it (how we missed it), 2022.
`https://blog.computationalcomplexity.org/2022/04/the-roeder-problem-was-solved-before-i.html`.

[8] W. Gasarch, D. Harris, and T. Harris. The Roeder seq problem was solved before I posted it (math), 2022.
`https://blog.computationalcomplexity.org/2022/04/the-roeder-seq-problems-was-solved.html`.

[9] W. Gasarch, E. Kaplitz, and E. Metz. Open Problems Column: How does $a_n = a_{n-1} + a_{\lfloor n/2 \rfloor}$ behave mod? *SIGACT News*, 53(3), 2022.
`https://www.cs.umd.edu/~gasarch/open/cseq.pdf`.

[10] W. I. Gasarch. Guest column: The third P=?NP poll. *SIGACT News*, 50(1):38–59, 2019.
`https://www.cs.umd.edu/users/gasarch/papers/poll3.pdf`.

[11] P. Kropitz. Busy beaver problem. Master's thesis, Charles University in Prague, 2011.
`https://is.cuni.cz/webapps/zzp/detail/49210/`.

[12] S. Lin and T. Radó. Computer studies of Turing machine problems. *Journal of the Association of Computing Machinery (JACM)*, 12(2):196–212, 1965.

[13] H. Marxen and J. Buntrock. Attacking the Busy Beaver 5. *Bulletin of the European association of theoretical computer science (BEATCS)*, 40:247–251, 1990.
`http://turbotm.de/~heiner/BB/mabu90.html`.

[14] P. Michel. Busy Beaver competition: a historical survey, 2019.
`https://arxiv.org/abs/0906.3749`.

[15] A. Pollington. There is a long simple path in the divisor graph. *Ars Combinatorics*, 16:303–304, 1983.

[16] C. Pomerance. On the longest simple graph in the divisor graph. *Congressus Numerantium*, pages 291–204, 1983.
`https://math.dartmouth.edu/~carlp/divisorgraph.pdf`.

[17] E. Saias. Etude du graphe divisoriel 5 (In French), 2021.
`https://arxiv.org/pdf/2107.03855`.

[18] Skelet. Skelet's 43 machines, 2003.
`https://bbchallenge.org/skelet`.

[19] T. Sterin. No name, 2016.
`https://bbchallenge.org/1RB1RA_0LC1LE_1LD1LC_1LA0LB_1LF1RE_---0RA`.

[20] G. Tenenbaum. Corrigendm et etude du graphe divisoriel. *Sur un probleme de crible et ses applications-2*, 28:291–204, 1995.

[21] W. van Doorn. On the congruence properties and growth rate of a recursively defined sequence, 2024.
`https://arxiv.org/pdf/2406.09532`.

[22] C. Website, 2024.
`https://bbchallenge.org/72006367`.

[23] Wikipedia. Coq (software).
https://en.wikipedia.org/wiki/Coq_(software).