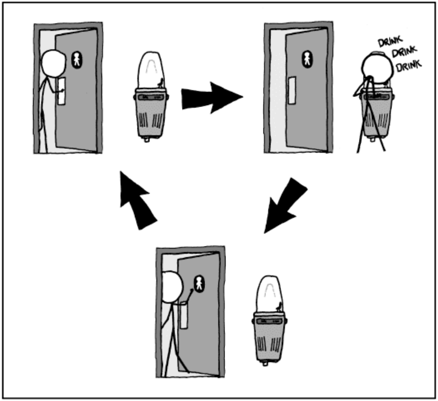
What are...Chaitin's constants?

Or: Computing a glimpse of randomness

Halting problems (HP)



I AVOID DRINKING FOUNTAINS OUTSIDE BATHROOMS BECAUSE I'M AFRAID OF GETTING TRAPPED IN A LOOP.

- **Problem** For a given program can one decide whether it halts or not?

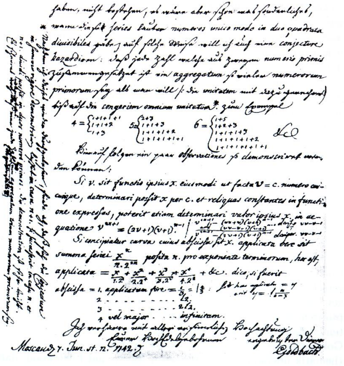
  Pseudocode examples

  does halt: print "Hello, world!"

  does not halt: while (true) continue

- **Warning** There is no general algorithm to do this

Solving “all” mathematical problems



- Goldbach's conjecture Every $n \geq 4$ is the sum of two primes
- Goldbach's program Write a program P that searches for counterexamples

$$\text{Goldbach's HP}: \quad \begin{array}{rcl} P \text{ halts} & \Rightarrow & \text{GC false} \\ P \text{ does not halt} & \Rightarrow & \text{GC true} \end{array}$$

- Many problems can be solved in the same way

Imagine the following

- Chaitin's omega Ω encodes whether programs will halt
- Knowing enough digits of Ω, one could calculate the HP for all programs
- Thus, all of mathematics turns into a digit hunt!?

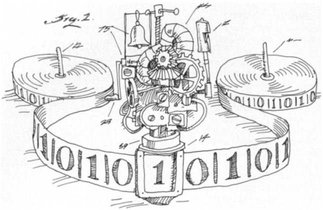
Enter, the theorem

p is a program expressed in binary form, U universal Turing/Chaitin machine (UTM)

$$\Omega_U = \sum_{p \text{ halts}} 2^{-|p|}$$

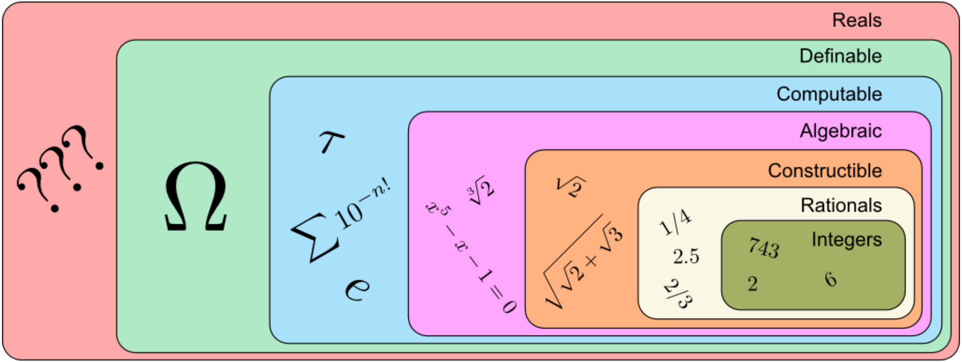
Then $\Omega_U \in [0, 1]$, the halting probability

- $\Omega_U \leftrightsquigarrow$ probability that a random program will halt
- Note that Ω_U depends on U
- Turing machine (TM) A mathematical model of computation



- UTM A TM that simulates an arbitrary TM on arbitrary input

Computing randomness



Knowing enough digits of Ω_U would "solve all problems", however:

- ZFC (if sound) can determine the value of only finitely many bits of Ω_U
- Algorithmically random To get n digits one needs a program of length $\approx n$
- Not computable No computable function enumerates Ω_U binary expansion

Thank you for your attention!

I hope that was of some help.