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

fation, might hopkofon , al mine about for mal forwardings ? , me singlet ferres lauter mumares unico modo in due operation withiles gut y suif filly a sign will of any nine conjecture hazadiom : Saf jato zaft welfe sit zuran numeris primi Anumgufitzat if an aggregation of vialow numerorung orver gling all when will for in white and see younglow +={:+:+2 5:+++3 +=2:++2 5:+++3 ++++++ Binned folgen vin gave offervationes of demonstrivent win Si v. sit functio ipsias x. cius mode at facta V = c. numbro an congres, determinari perfit & por c. et reliques constances in funch an determinari val

• Goldbach's conjecture Every  $n \ge 4$  is the sum of two primes

Goldbach's program Write a program P that searches for counterexamples

Goldbach's HP : P halts  $\Rightarrow$  GC false P does not halt  $\Rightarrow$  GC true

Many problems can be solved in the same way

## Imagine the following



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

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

- $\blacktriangleright \ \Omega_U \iff$  probability that a random program will halt
- ► Note that  $\Omega_U$  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  $\Omega_U$  would "solve all problems", however:

- ► ZFC (if sound) can determine the value of only finitely many bits of  $\Omega_U$
- Algorithmically random To get *n* digits one needs a program of length  $\approx n$

Not computable No computable function enumerates  $\Omega_U$  binary expansion

Thank you for your attention!

I hope that was of some help.