Het Halting Probleem
en
Post Correspondence Probleem
2024

Formulering

Bestaat er een programma (TM, programma voor universele TM, of programma in Python, etc) `Halt` dat gegeven een willekeurig ander programma `P` (evt met input `I`) voor dit programma aangeeft of dit programma stopt.

Als dit zou bestaan, dan kunnen we een groot aantal open wiskundeproblemen makkelijk oplossen.

Neem bv het Goldbach vermoeden:
Ieder even getal groter dan 2 is te schrijven als de som van 2 priemgetallen.

Dit is een beroemd (nog) onbewezen vermoeden!

Goldbach

```
import math

def isPrime(n):
    wn = math.sqrt(n)
    wn = math.trunc(wn)
    for k in range(2,wn+1):
        if n % k == 0:
            return False
    return True


def isSum2Primes(n):
    for k in range(2,n//2+1):
        if isPrime(k) and isPrime(n-k):
            return True
    return False
```

```
def goldbach_bounded(b):
    # every even number is the sum of 2 primes
    # program
    n = 4
    while n < b:
        if not isSum2Primes(n):
            return False
        n += 2
    return True

def goldbach():
    # every even number is the sum of 2 primes
    # program
    n = 4
    while True:
        if not isSum2Primes(n):
            return False
        n += 2
    return True
```

goldbach() stopt alleen als als er een even getal is, dat niet de som is van priemgetallen

Goldbach

Als `Halt` bestaat dat kunnen we dus bepalen of `goldbach()` wel of niet stopt en kunnen we dus het Goldbach vermoeden bewijzen/ontkrachten.

Het programma `Halt` zal dus wel niet kunnen bestaan!

Er volgt nu een schets van een bewijs

- geen tentamenstof
- wel het idee

Paradoxen

In het bewijs creëren we een paradox vergelijkbaar met de volgende uitspraken:

Deze zin is niet waar!

Beroemd voorbeeld
Iemand is ter dood veroordeeld, maar mag zelf kiezen of hij wordt onthoofd of opgehangen. Hij mag een uitspraak doet. Als de uitspraak waar is wordt hij onthoofd. Als hij niet waar is wordt hij opgehangen.

Wat zou jij zeggen?

Paradoxen

Beroemd voorbeeld
Iemand is ter dood veroordeeld, maar mag zelf kiezen of hij wordt onthoofd of opgehangen. Hij mag een uitspraak doet. Als de uitspraak waar is wordt hij onthoofd. Als hij niet waar is wordt hij opgehangen.

Wat zou jij zeggen?

Ik word opgehangen!

Als hij wordt opgehangen had hij onthoofd moeten worden

Bewijs van het Halting: Hulpfuncties

We introduceren 2 hulpfuncties, loop en stop:

```
def loop():
   a = 3
   while a > 0:
      a = (a + 1) - 1

def stop():
   return 3
```

loop termineert duidelijk niet en stop duidelijk wel!

Bewijs van Halting: `halt` functie

Het bewijs is uit het ongerijmde. Dus we nemen aan dat de functie `halt` wel bestaat en we proberen tot een tegenspraak te komen. `halt` heeft dus de volgende eigenschap:

```
def halt(f):
   if f() stopt:
      return True
   else:
      return False
```

Bewijs Halting: de paradox!

Definieer nu:

```
def p():
   if halt(p):
      loop()
   else:
      stop()
```

Let op: p is recursief (roept zichzelf aan in de definitie)!
Als `halt(p)` True is, dan eindigt `p()` niet,
maar als `halt(p)` False, dan eindigt `p()` wel.
We hebben dus een tegenspraak vergelijkbaar met de leugenparadox!
Conclusie: `halt` kan klaarblijkelijk niet bestaan!

Opmerking

Dit geeft wel de essentie van het bewijs, maar de details rammelen.

We moeten strikt genomen onderscheid maken tussen een functie `f` en zijn representatie in Python als datastructuur (noem deze fr).

`Halt` moet eigenlijk `fr` als argument krijgen. Dit compliceert de definitie van `p` wel, maar maakt het niet onmogelijk!

Waarom dit onderscheid?
Dit heeft te maken met het feit dat je in de executie-omgeving van een programmeertaal niet in een programma (functie) zelf kunt kijken (je kan de functie eigenlijk alleen maar uitvoeren).
Als `Halt` echt zou bestaan, kan dit alleen maar omdat `Halt` aan de structuur (representatie) van een functie kan zien of deze wel of niet termineert!

Post Correspondence Probleem

Eenvoudig voorbeeld van een onbeslisbaar probleem
(probleem waar geen algoritme voor kan bestaan)
Domino steentjes met boven en onder een string.
Vind een ordening van de steentjes zo dat boven en onder dezelfde string staat of zeg anders dat het onmogelijk is.
Hierbij mogen steentjes meerdere keren gebruikt worden!

Zie beschrijving wikipedia

In het bewijs laat men zien dat het simuleren van een willekeurige Turingmachine gecodeerd kan worden mbv domino steentjes en dat het vinden van een correspondentie neer komt op de oplossing van het Halting probleem. Aangezien dat niet mogelijk is, is het ook niet mogelijk iha het Post probleem op te lossen!

Diophantische Vergelijkingen

Beschouw vergelijkingen van het type:

$a*x^n + b*y^m + c*z^k = 0$

met a b c n m k gehele getallen.

Bv $x^3 + y^3 + z^3 = 42$

Bestaan er een oplossing met gehele getallen x,y,z?

Nog onbekend voor 42 (geldt wel voor andere n < 100).

Dit probleem is in het algemeen onbeslisbaar.

Diophantische Vergelijkingen

Er kan geen methode bestaan die voor een willekeurige Diophantische vergelijking kan zeggen of deze een oplossing heeft (Negatief antwoord op Hilberts 10e probleem)