Gödels theorem — An Incomplete Guide to Its Use and Abuse, Hoofdstuk 3

Koen Rutten, Aris van Dijk

30 mei 2007

Inhoudsopgave

1 Verzamelingen

1.1 Definitie

```
Computably Enumerable = Recursively Enumerable = Effective Enumerable
= '(berekenbaar) opsombaar'

Computably Decidable = Computable = Decidable
= '(berekenbaar) beslisbaar'
```

opsombaar een programma dat alles in E print (merk op: hoeft niet te eindigen (E kan ook oneindig zijn), mag meer keren dezelfde uitprinten)

beslisbaar een programma dat gegeven een invoer x bepaalt of x in E is

- Beslisbaar heeft twee betekenissen:
  - Als eigenschap van een verzameling E: Kan van elke x beslist worden of $x \in E$
  - Als eigenschap van een zin A in een formeel systeem S: is er een bewijs voor A of $\neg A$ in S
- 'Berekenbaar' heeft ook een intuïtieve betekenis, maar hier gaat het over Turing berekenbaar.

1.2 Eigenschappen

Eigenschap 1.1. *Als A en B opsombaar zijn, is $A \cup B$ ook opsombaar.*

Er zijn dus programma's om A en B op te sommen. Maak een programma dat om de beurt een stap van het ene en het andere programma uitvoert.

Eigenschap 1.2. *E beslisbaar $\Rightarrow$ E opsombaar*

Dit is in te zien door de volgende procedure te hanteren: loop *alle* zinnen af, en printen alleen die zinnen die in E zijn (beslisbaar!).

Er geldt op deze manier niet: opsombaar $\Rightarrow$ beslisbaar, want het kan (op een opsommanier) oneindig lang duren voordat je kunt zeggen dat een zin x niet in E zit

N.B. Dit is geen bewijs, want misschien is er wel een slimmere manier dan de opsommanier. Het bewijs is gegeven door de Church-Turing stelling, zie hieronder.

Eigenschap 1.3. *E beslisbaar $\iff$ E opsombaar en complement van E opsombaar*

De beslissingsprocedure kan dan namelijk geschreven worden door om de beurt uit E en het complement van E af te tellen. Dan duurt het dus eindig lang voordat je kunt zeggen dat een zin x in E of complement E zit.

Andersom, als E beslisbaar is, geldt dat het complement van E ook beslisbaar is. En uit beslisbaarheid volgt opsombaarheid, zie Eigenschap 1.2.

1.3 Opsombaar maar onbeslisbaar

1.3.1 Turings verzameling niet-stoppende programma's

De onbeslisbaarheidsstelling van Church-Turing bewijst dat er opsombare verzamelingen zijn die niet beslisbaar zijn. Hiervoor gebruikte Turing een bewijs dat gaat over een eigenschap van programma's.

Een *programma* bestaat uit: statements (opdrachtregels), input(invoer), output (uitvoer). Programma's kunnen oneindig lang doorgaan, en nooit stoppen (to terminate / to halt). Turing construeerde een bewijs dat het niet beslisbaar is of een programma ooit stopt.

We kijken hier naar programma's met één invoer, en die nooit stoppen of stoppen met éé uitvoer.

1. Kies een programmeertaal (Turing-compleet: kan alles berekenen).
2. Geef alle programma's P_i een index i.
3. Met $P_i(m)$ noteren we dat het programma met index i stopt als het invoer m krijgt.
4. Definieer K als de verzameling van indexen van programma's, die als ze hun eigen index als invoer krijgen, stoppen: $K = \{i | P_i(i)\}$
5. Zie in dat K opsombaar is.
   (a) Met $P_i^n(m)$ noteren we dat het programma met index i stopt in hooguit n stappen als het invoer m krijgt. Dit is een berekenbare eigenschap.
   (b) De verzameling van alle paren $\langle i, n\rangle$ is opsombaar, en dus ook de verzameling $\{\langle i, n\rangle | P_i^n(i)\}$.
   (c) Nemen we alle i's uit deze verzameling, dan krijgen we K.
6. Stel dat K beslisbaar is.
   (a) Maak een programma P:

   invoer $\leftarrow i$
   if $i \in K$ **then**
   geef de uitvoer van P_i (met i als invoer), en plak er een symbool achter (P_i stopt, omdat de index i van P_i in K is)
   else
   geef uitvoer 0, stop.
   end if

   P is ook een programma met een index, zeg m.
   (b) Geef P zijn eigen index m, dan gebeurd het volgende:

   invoer $\leftarrow m$
   if $m \in K$ **then**

geef de uitvoer van P_m (met m als invoer), en plak er een symbool achter

end if

(c) Wat raar! P_m en P geven verschillende uitvoer met m als invoer, want P plakt er een symbool achter.

7. Dus K is niet beslisbaar.

1.3.2 Diagonaalbewijs

Cantor en diagonaal Van Cantor kennen we het diagonaalbewijs dat een machtsverzameling van een verzameling A, $\wp(A)$, altijd groter is dan A, zelfs voor oneindige A. Dit wordt bewezen met een tegenspraak: We nemen aan dat de elementen uit van $\wp(A)$ te tellen zijn met de elementen van A. Om dat te proberen beginnen we een lijst van alle elementen van $\wp(A)$, dat zijn dus deelverzamelingen van A. We kunnen een deelverzameling van A noteren als een opsomming van van waar en onwaar, die van elk element van A aangeeft of het er in zit.

$$R = \begin{pmatrix} A_0 \to & \bot & \top & \bot & \top & \cdots \\ A_1 \to & \bot & \bot & \top & \bot & \cdots \\ A_2 \to & \top & \bot & \top & \top & \cdots \\ A_3 \to & \top & \top & \top & \top & \cdots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \ddots \end{pmatrix} \tag{1}$$

We zijn met het opsommen begonnen $(A_0, A_1, \ldots)$. Maar Cantor liet zien dat het onbegonnen werk is: er is altijd een rijtje dat niet afgeteld is[1]: dat is de rij P die je krijgt als je de diagonaal van deze tabel (dus $\bot, \bot, \top, \top, \ldots$) 'omflipt' (dus $\top, \top, \bot, \bot, \ldots$)! Die verschilt namelijk van elke rij n ten minste op de n-de plek.

Algemeen Nu de generalisatie. De tabel, die we R genoemd hebben, is in het geval van Cantor te omschrijven als: $R_{a,b}$ = element a van $\wp(A)$ bevat het b-de element van A. De diagonaalrij P is te omschrijven als $P_a = \neg R_{a,a}$. Nu volgt het bewijs: $P \neq R_a$ voor elke a, waar we met R_a de a-de rij van R bedoelen.

Nu Turings voorbeeld. $R_{a,b}$ = de uitvoer die een gestopt programma P_a (met index a) geeft bij invoer b. (R is nu een functie, maar dat maakt niet zoveel uit.) Definiëer nu programma P zodanig dat $P_a = f(R_{a,a})$, met f een functie waarvoor geld $\forall x : f(x) \neq x$. P is dus het programma dat met invoer van zijn eigen index een gewijzigde uitvoer geeft! Nu volgt: $P \neq P_m$ voor elke m.

[1] Er zijn er eigenlijk zelfs oneindig veel niet afgeteld.

1.3.3 Posts simpele verzamelingen

Posts opsombare 'simpele verzamelingen' (dat ze bestaan komt later) zijn extreem onbeslisbaar, want niet alleen kan het complement van die verzamelingen niet opgesomd worden, maar zelfs elke oneindige deelverzameling van dat complement is opsombaar.

(Turings K is niet simpel, want het complement van K (de programma's die *niet* eindigen), heeft een oneindige deelverzameling, die opsombaar is (makkelijk te genereren).)

1.3.4 Oplossingen voor Diophantische vergelijkingen (Hilberts probleem 10, de Matiyasevich-Robinson-Davis-Putnam (MRDP) stelling)

Hilberts Probleem 10 was: vind een algoritme (rekenprocedure) dat alle Diophantische vergelijkingen kan oplossen. De MRDP-stelling liet echter zien dat zo'n algoritme niet bestaat.

Zie eerst in dat de verzameling oplossingen van de Diophantische vergelijkingen opsombaar is. Beweringen (waar of onwaar) van de vorm "$D(x_1, \ldots, x_n) = 0$ heeft een oplossing $x_1 = k_1, \ldots, x_n = k_n$" zijn opsombaar (want ze zijn beslisbaar, je kunt zeggen of een zin die vorm heeft of niet). Om de *ware* oplossingen op te sommen, lopen we gewoon deze verzameling beweringen af en kijken we of de vergelijking klopt.

Op een zelfde manier geldt het ook voor geparametriseerde Diophantische vergelijkingen. Namelijk, voor een $D(x_1, \ldots, x_n, y) = 0$ kunnen we ook alle getallen k voor y die de vergelijking oplossen, opsommen; laten we die verzameling van k's even E_D noemen, een "Diophantische" verzameling voor D.

De MRDP-stelling bewijst nu iets onverwachts: *elke* opsombare verzameling van getallen is een "Diophantische" verzameling, dus een set oplossingen voor een Diophantische vergelijking! Dat heeft de volgende consequenties:

- Omdat er opsombare verzamelingen natuurlijke getallen zijn die onbeslisbaar zijn (zie bijvoorbeeld de Church-Turing-stelling), kan er geen algoritme zijn dat voor elke Diophantische vergelijking kan beslissen of die vergelijking een oplossing heeft. Neem bijvoorbeeld een onbeslisbare K: dan y is in $K \iff D(x_1, \ldots, x_n, y) = 0$ een oplossing heeft.

- De verzameling van *niet-oplosbare* Diophantische vergelijkingen is niet opsombaar. (Volgens Eigenschap 1.1.)

- Er is nog een consequentie, die het gevolg is van het feit dat de MRDP-stelling bewezen kan worden in elk systeem met voldoende rekenkunde (zoals PA).

  Volgens MRDP geldt voor elke stelling A: $A \iff D(x_1, \ldots) = 0$ geen oplossing heeft!

  In het bijzonder: er kan ook zo'n vergelijking gevonden worden voor de bewering "PA is consistent"!

2 Formeel systeem

2.1 Definitie

Formeel systeem $F = \langle L, R, A \rangle$, waarbij L = taal, R = inferentieregels, A = axioma's. $A + R \Rightarrow$ stellingen T

2.2 Eigenschappen

Eigenschap 2.1. *T moet opsombaar zijn*

N.B. Als in een systeem de *bewijzen* beslisbaar zijn (dat lijkt vanzelfsprekend!), dus als we T willen opsommen kunnen we de bewijzen opsommen en daarvan de laatste regel printen.

N.B. De bewijzen hoeven niet beslisbaar te zijn, de opsombaarheid van de bewijzen is genoeg. Zoeken voor een bewijs is dus mogelijk in eindige tijd.

Eigenschap 2.2. *Als F* volledig *is (dus als ten minste de eerste onvolledigheidsstelling niet geldt voor dit systeem!), is T beslisbaar. Ook wel genoemd: "F is beslisbaar".*

Bewijs: of F is inconsistent, dan is T gelijk aan alle zinnen in taal L van F, of F is consistent, dan kunnen we in eindige tijd voor elke zin A de opsombare verzameling T afgaan tot we A of $\neg A$ als stelling tegenkomen.

N.B. Er zijn voorbeelden F die beslisbaar (van elke A kan gezegd worden of A in T is) zijn zonder volledig (dat A of $\neg A$ in T is) te zijn! Bijvoorbeeld: een formeel systeem zonder axioma's!

2.3 Een alternatief bewijs voor onvolledigheid

1. Neem een *opsombare, maar niet beslisbare* verzameling E (zoals de Diophantische vergelijkingen die oplossingen hebben). Het complement van E zal dan *niet opsombaar* zijn (volgens Eigenschap 1.3).

2. Neem een Σ-correct formeel systeem S. S zal dan geen onware Goldbachachtige uitingen bewijzen.

3. Laat B de verzameling zijn van *ware* uitspraken van de vorm "x is niet in E" (dit is een verzameling even groot als het complement van E).

4. Laat A de verzameling zijn van alle *in S bewijsbare* stellingen van de vorm "x is niet in E". (A is dus opsombaar, zie Eigenschap 2.1.)

5. A is dus een deelverzameling van B, maar A is, in tegenstelling tot B, opsombaar. Dus $B \setminus A$ is niet leeg, en zelfs niet opsombaar (Anders zou $A \cup (B \setminus A) = B$) dat ook zijn, zie Eigenschap 1.1.)

6. Er zijn dus ware zinnen van de vorm "x is niet in E" waarvoor geen bewijs, noch een tegenbewijs bestaat in S: S is onvolledig.

We hebben hier gebruikt dat in S de stellingen van de vorm "x is in E" zijn te formuleren. Gebruiken we voor E bijvoorbeeld de Diophantische vergelijkingen die geen oplossingen hebben, dan hebben we alleen een beetje rekenkunde nodig. Als we het voor *elke* E willen doen, kunnen we een aritmetisering van de taal van computatie formuleren (er is een model van computatie, zoals Turings machine, nodig).

Wat voetnoten:

1. In het speciale geval van *onoplosbare* Diophantische vergelijkingen kan geen enkel consistent systeem alle beweringen van de vorm "$D(x_1, \ldots, x_n) = 0$ heeft een oplossing $x_1 = k_1, \ldots, x_n = k_n$" beslissen, want de onbeslisbare Rosser zin is equivalent aan zo'n bewering.

2. Als E een *simpele* (Posts) verzameling is, kan S maar *eindig* veel van de beweringen "k is *niet* in E" bewijzen, ook al zijn er oneindig veel ware beweringen van die vorm.

2.4 Essentiële onbeslisbaarheid

We hebben aan het begin van de vorige sectie gezien dat een Σ-correct systeem S met voldoende rekenkunde om oplossingen voor Diophantische vergelijkingen te bewijzen (zoals PA), onbeslisbaar is (d.w.z. de stellingen T van S onbeslisbaar zijn).

Maar met hetzelfde argument volgt ook dat alle Σ-correcte systemen die PA bevatten, onbeslisbaar zijn. Met een variant op dit bewijs kan bewezen worden dat elke *consistente* (dus niet persé Σ-correcte) extensie van PA onbeslisbaar is. PA wordt daarom *essentieel onbeslisbaar* genoemd. Daaruit volgt ook dat ze onvolledig zijn (zie Eigenschap 2.2).

2.4.1 Nieuwe bewijzen

Wat gebeurt er als je een systeem S uitbreidt met een nieuw axioma? Er zullen nieuwe stellingen M te bewijzen zijn. We bezien eigenschappen van M.

1. Stel S is essentieel onbeslisbaar, en A is een onbeslisbare zin in S.

2. Maak een systeem $S' = S + A$.

3. Definieer de verzameling M als alle stellingen van S' die niet stellingen van S zijn.

4. M is *niet opsombaar*.

   (a) Beschouw de zin "$A \vee B$". Voor elke B is dit een stelling in S', want A is een axioma in S'.

   (b) Die zin is dan en slechts dan in M als die zin geen stelling is in S. Omdat geldt: $S \vdash A \vee B \iff S \vdash \neg A \to B \iff S + \neg A \vdash B$, is die zin in M dan en slechts dan als B niet een stelling is van $S'' = S + \neg A$.

(c) Zou M opsombaar zijn, dan zijn de niet-stellingen van S'' opsombaar (want we kunnen tijdens het opsommen van M beslissen alleen gevallen van de vorm $A \vee B$ te printen). En omdat de stellingen van S'' per definitie opsombaar zijn, is S'' beslisbaar.

(d) En dat is inconsistent met onze aanname dat elke consistente extensie van S onbeslisbaar is; S'' ís consistent, want A is onbeslisbaar in S.

5. Dus in S' is een rijke (onopsombare) verzameling M van nieuwe stellingen!

2.4.2 Kortere bewijzen

Levert het systeem S' ons ook *kortere bewijzen* voor stellingen die al in S zijn? Antwoord: ja, met het volgende bewijs.

N.B. Een 'korter bewijs' betekent hier een bewijs met een kleiner Gödel-getal.

1. Neem een *willekeurige* berekenbare functie f over natuurlijke getallen, nu kan bewezen worden dat er een stelling A in S is waarvan de lengte p van het kortste bewijs voor A in S groter is dan $f(p')$, dus de functie f toegepast op de lengte p' van het kortste bewijs voor A in S'. (Voorbeelden: $p > 10^{1000} \times p'$ of $p > 2^{p'} \ldots$)

2. Want stel dat het niet zo zou zijn, dan zou het volgende mogelijk zijn:

   (a) Maak een opsomming van de stellingen in S'.

   (b) Voor elke bewijs van A in S', bepaal de lengte p', en genereer alle bewijzen in S met lengte maximaal $p = f(p')$.

   (c) Kom je geen bewijs (in S) tegen voor A (dan is er dus geen bewijs), output dan A.

3. Daarmee hebben we een opsomming voor de stellingen van S' die geen stelling zijn in S, maar dat is in strijd met de eigenschap van essentiële onbeslisbaarheid.

4. Er bestaat dus geen berekenbare functie f met die eigenschap.