Kunnen wij elke machine verslaan? Beschouwingen rondom Lucas' Argument*

Albert Visser†

Department of Philosophy, Utrecht University

Heidelberglaan 8, 3584 CS Utrecht, The Netherlands

email: Albert.Visser@phil.uu.nl

24 oktober 2004

Samenvatting

Hoe mooi zou het niet zijn als we kort en overtuigend konden aantonen dat we in tenminste één opzicht superieur zijn aan machines. J.R. Lucas dacht met behulp van Gödels Onvolledigheidsstellingen zo'n argument te kunnen produceren. Helaas, blijken er enkele haken en ogen aan zijn redenering te zitten. Dit artikel geeft een expositie van Lucas' argument. Na beschouwing van enige bezwaren, kom ik tot de conclusie dat Lucas' argument een drogreden is.

Key words: Gödel's Theorem, mind, machine, mechanism, artificial intelligence

1 Inleiding

Weinig zaken zijn fascinerender dan een kort, op het eerste gezicht eenvoudig, schijnbaar waterdicht argument met een grootse conclusie. Het mooiste voorbeeld is nog steeds Anselmus' Godsbewijs. Dat gaat ongeveer zo:

> De dwaas zei in zijn hart: er is geen God. Maar zelfs de dwaas zal moeten toegeven dat hij zich iets kan voorstellen zodat niets groter is dan dat *iets*. Maar als dit iets niet echt, maar alleen in het verstand bestaat, kunnen we ons iets nog groters voorstellen: namelijk iets

*Dit artikel is een licht verbeterde versie van [Vis86].

†Ik dank de redacteuren van *Geest, Compute, Kunst*, 1986, Peter Hagoort en Rob Maessen, voor hun gedetailleerde kritiek op de eerdere versie van dit artikel. Erik Krabbe, Fer-Jan de Vries stelden vele verbeteringen voor in de eerste versie. Heb dank, Erik en Fer-Jan! Lev Beklemishev, Leon Horsten en Menno Lievers becommentarieerden de tweede versie. Dank Lev, Leon en Menno! Ik ben erkentelijk aan Leon Horsten en Allard Tamminga die publlicatie van de huidige versie mogelijk maakten. Ik dank de stichting Grafiet voor de welwillende toestemming om het artikel opnieuw te publiceren.

> dat óók nog bestaat. We hebben nu duidelijk een tegenspraak. Dus bestaat er echt iets zodat er niets groters gedacht kan worden. Dit iets noemen we God ...

Het gekke is dat dit argument, hoe mooi ook, geen greintje overtuigingskracht heeft —voor mij niet, tenminste. Ook al zou ik geen flauw idee hebben wat er mis is, dan nog zou de conclusie mij na het horen van het argument absoluut niet waarschijnlijker voorkomen. Waarom? Omdat ik de intuïtie heb dat *zo'n soort* stelling niet bewezen kan worden met zo'n simpel 'logisch' argument. Er worden heel weinig specifieke inzichten aangaande God gebruikt, terwijl Hij —zo hij al bestond— toch een heel individueel Iets zou moeten zijn. Anselmus laat God meer lijken op het neutrino, waarvan het bestaan voorspeld kon worden nog voor het experimenteel ontdekt was.

In dit artikel wil ik niet Anselmus' Argument bespreken maar dat van Lucas, J.R. Lucas om precies te zijn. Lucas' Argument is gericht tegen de dwaas die in zijn hart zei: de mens is een machine. Het argument is kort, maakt gebruik van éénvoudige, maar diepe wiskundige stellingen —de Onvolledigheidsstellingen van Gödel— en lijkt op weinig veronderstellingen te berusten. Lucas laat zien, of pretendeert te laten zien, dat er ten minste één ding is waarin wij elke machine de baas zijn: het Gödeliseren. Mijn eerste reactie op Lucas' Argument is, net als bij Anselmus, ongeloof dat zo'n soort argument zo'n soort conclusie zou kunnen bewijzen. Er wordt een wiskundige stelling, die betrekking heeft op precies omschreven objecten, toegepast op een wijdse, verwarrende, slechts vaag omlijnde problematiek. Het toepassen van wiskunde in de techniek is al een verfijnde kunst, die de meeste mensen alleen met veel oefening onder de knie krijgen; het toepassen van wiskunde in de filosofie is nog veel moeilijker ...

Zoals we verderop zullen zien neemt nadere beschouwing van Lucas' Argument deze eerste indruk niet weg. Integendeel. Waarom is het dan toch de moeite waard om hier over dit argument te lezen? Wat maakt het tot meer dan een historische curiositeit? Ik denk dat het inzicht dat zulke argumenten niet werken zelf belangrijk is. Om dit uit te leggen moet ik eerst iets zeggen over de Onvolledigheidsstellingen van Gödel. Het verschijnen van de Gödelstellingen in 1930 was een culturele gebeurtenis van de eerste orde. De stellingen toonden aan dat Hilberts formalistische programma voor het funderen van de wiskunde niet uitvoerbaar was. Ze gaven een impuls aan de ontwikkeling van begrippen als *mechanische rekenprocedure* (*algorithme*) en *formeel systeem*. Ook het esthetisch oogpunt is hier belangrijk: de bewijzen zijn wonderschoon. Door combinatie van een aantal in wezen simpele ideeën worden diepe resultaten bereikt. Er bestond wijd en zijd de indruk dat de Gödelstellingen veel verdere en diepere implicaties hadden. Ze zouden essentiële beperkingen van machines, van het wiskundig denken, van de menselijke geest aantonen.[1] Deze drie beperkingen zijn heel verschillend en staan zelfs min of meer tegenover elkaar. In welk soort beperking men geloofde was afhankelijk van bepaalde vooropgezette ideeën. Tegenwoordig lijken dit soort 'implicaties' van de Gödelstellingen minder plausibel dan voorheen. Waarom? In de eerste plaats zijn zulke indrukken

[1] Gödels eigen ideeën hieromtrent zijn te vinden in [Wan74], p. 324-328.

nu eenmaal onderhevig aan culturele slijtage. In de tweede plaats weten wij nu veel meer en dat werkt altijd ontnuchterend: de Gödelstellingen zijn voor ons gewoon wiskundige stellingen geworden met allerlei generalisaties, beperkende clausules enzovoorts. In de derde plaats hebben filosofen zoals Lucas getracht deze 'implicaties' te beargumenteren: wij kunnen nu in detail zien waarom deze argumentatie niet werkt. Lucas heeft het vage ongearticuleerde idee dat de Gödelstellingen laten zien dat de menselijke geest niet machinaal is, concreet gestalte gegeven in zijn Argument. Het is belangrijk de onjuistheid van Lucas' Argument in te zien, omdat het analyseren van dat Argument het ons mogelijk maakt afstand te nemen van het idee dat de Gödelstellingen een substantieel verschil tussen mensen en machines aantonen. Hierdoor krijgen we een beter zicht, op zowel de Gödelstellingen als de mens-machine problematiek.[2,3]

De verdere opzet van dit artikel is als volgt. Sectie 2 is een korte introductie in de mens-machine problematiek. In Sectie 3 verschaf ik het benodigde achtergrond materiaal aangaande de Gödelstellingen. Sectie 4 bevat tenslotte een expositie en bespreking van Lucas' Argument.

2 Rondom de mens-machine problematiek

Lucas' Argument richt zich tegen een mechanistische visie op de mens. In deze sectie wil ik in het kort beschrijven wat het Mechanisme behelst en ook wat het niet behelst. In het bijzonder wil ik duidelijk maken dat het Mechanisme een andere doctrine is dan het Fysicalisme. Fysicalisme en Mechanisme zijn beide vormen van reductionisme, maar wel verschillende vormen. Het inzien van het verschil is niet alleen van belang op 'hygiënische' gronden, maar ook om de volgende reden: *het Fysicalisme is een metafysische doctrine die voor veel beoefenaars van de moderne wetenschap prima facie plausibel is. De naïeve verwarring tussen mechanistische en fysicalistische thesen maakt dat antimechanistische claims al bij voorbaat iets verdachts krijgen. Ze lijken zich immers te richten tegen iets dat al als evident geaccepteerd is: het Fysicalisme.* Lucas' Argument richt zich duidelijk tegen het Mechanisme, of het ook antifysicalistische implicaties heeft, zo het al implicaties heeft, is niet direct duidelijk.

2.1 Het Mechanisme

Een rechtgeaarde Mechanist gelooft in een representatieve greep uit de volgende groep van thesen (en misschien ook nog wel in enkele nauw verwante thesen).

These 2.1 *Mensen zijn machines.*

[2]Ik geef graag toe dat deze bewering ietwat overdreven is. Men kan natuurlijk willekeurig veel "Lucas-argumenten" verzinnen. Het is onrealistisch van één argumentatiestrategie te verwachten dat zij alle denkbare varianten in één klap weerlegt. Filosofische discussies zijn nooit echt afgesloten.

[3]Het is interessant de Gödelstellingen te vergelijken met Heisenbergs Onzekerheidsrelatie. Het is belangrijk om in te zien dat Heisenbergs resultaat betrekkelijk weinig impliceert aangaande, zeg, de menselijke vrijheid.

Wel, wat zijn machines? Het is moeilijk je een definitie voor te stellen die niet zo begint: "Machines zijn artefacten die ...". Mensen zijn natuurlijk geen artefacten, ergo mensen zijn geen machines. Dit laatste is natuurlijk wel een heel flauw argument. Het gaat de Mechanist niet om het ontstaans aspect maar om bepaalde substantiële overeenkomsten tussen mensen en machines. We kunnen alleen maar concluderen dat These 2.1 in deze ongenuanceerde vorm onacceptabel is.[4] We kunnen water bij de wijn doen en bijvoorbeeld uitwijken naar een these als:

These 2.2 *Machinale modellen zijn adequaat voor de menselijke psychologie.*[5]

In de literatuur beperkt men zich vaak tot specifiekere thesen die betrekking hebben op één bepaald menselijk vermogen, zoals:

These 2.3 *Het menselijk denken is mechanisch.*[6]

Lucas' Argument richt zich het meest direct tegen deze laatste these. Het spreekt vanzelf dat bij een goede mechanistische theorie een uitleg hoort van wat 'mechanisch' nu eigenlijk is, met andere woorden: van wat nu precies de relevante overeenkomsten tussen menselijk denken en (zekere) machinale processen zijn. Bijvoorbeeld, het Functionalisme is niet slechts een theorie over de menselijke geest, maar ook over de aard van machinale toestanden en processen; er is sprake van discrete toestanden, die functioneel van aard zijn, en van stapsgewijs verlopende processen.[7]

Er zijn natuurlijk ook mechanistische thesen waarin in de eerste plaats iets beweerd wordt over machines en niet over mensen:

These 2.4 *Denkende machines zijn mogelijk.*

These 2.5 *Het is denkbaar dat een machine tevens een persoon is.*

These 2.6 *Machines kunnen bewustzijn hebben.*

These 2.7 *Machines kunnen een geest hebben.*

Een voordeel van de machine-gecentreerde thesen is dat erin duidelijker de mogelijkheid wordt opengelaten dat eigenschappen als bewustzijn aanwezig zijn, zonder dat er sprake is van een substantiële gelijkenis met mensen. Ook Martianen zijn wellicht bewuste, denkende personen, maar ze hoeven geen goede modellen te zijn van de menselijke psychologie.

[4] Het probleem hier is analoog met de vraag of de genetische code wel echt een code is. Wat is de gecodeerde boodschap? Wie of wat is de zender? Wie is de ontvanger? Het belangrijke punt is dat het er niet zoveel toe doet of de genetische code een code is of niet; waar het om gaat is dat het voor veel doeleinden verhelderend is de genetische code te zien als code.

[5] These 2.1 verwoordt een vorm van *ontologisch reductionisme*. These 2.2 daarentegen belichaamt een vorm van reductionisme *dat betrekking heeft op verklaringen*.

[6] 'Mechanisch' moet hier geïnterpreteerd worden als: gelijkenis vertonend met machinale processen. Niet als: machinaal. De claim dat het menselijk denken machinaal is, stuit op hetzelfde bezwaar als These 2.1.

[7] Voor een leesbare introductie tot het Functionalisme, zie [Fod81].

2.2 Het Fysicalisme

Fysicalisme kent vele varianten. Deze hebben alle gemeen dat er een niveau van werkelijkheid —het 'fysische'— geponeerd wordt met een speciale status. In reductionistische varianten wordt deze speciale status als volgt uitgelegd: alle andere —wellicht: schijnbaar andere— niveaus van werkelijkheid kunnen gereduceerd worden tot het fysische niveau. Wat 'reductie' hier inhoudt kan nog op vele manieren ingevuld worden. Laat ik om de discussie niet te veel te compliceren een liberale, niet reductionistische vorm van Fysicalisme schetsen. De theorie ziet er zo uit:

a. Er is een niveau van werkelijkheid, het 'fysische', dat causaal gesloten is. Dat wil zoiets zeggen als dat gebeurtenissen op dat niveau gebeurtenissen op andere niveaus niet nodig hebben om plaats te vinden.[8]

b. Het fysische is datgene wat bestudeerd wordt in de fysica.

c. Fysische objecten zijn bijvoorbeeld electronen, electro-magnetische velden en zwarte gaten. Van alles is echter niet-fysisch.

d. Op het fysische niveau komen zaken als functie en betekenis niet voor. Deze stoel waarop ik zit, is geen puur fysisch object. De stoel heeft een functie: hij is om op te zitten. Als hij die functie niet had, zou het geen stoel zijn. Fysische objecten hebben in die zin geen functie. Een stoel kan kapot gaan; fysische objecten gaan niet kapot, ze houden hoogstens op te bestaan. Natuurlijk heeft de stoel wel een fysische drager.[9]

e. Veroorzaking wordt gemedieerd door het fysisch niveau. Bijvoorbeeld de gedachten van voorzitter Mao veroorzaakten (mede) de Culturele Revolutie. Maar noch de gedachten van Mao, noch de Culturele Revolutie zijn fysische objecten. De bewering is dat deze veroorzaking niet kon plaats vinden zonder fysische belichaming van die gedachten, op papier, in geluidsgolven, in electro-magnetische golven.[10]

[8] Het hier beschreven reductionisme is weer ontologisch. Het is niet moeilijk een variant te bedenken die betrekking heeft op verklaringen.

[9] Een problematisch aspect van deze doctrine is dat zij toelaat dat verschillende objecten op dezelfde tijd op dezelfde plaats kunnen zijn. Dit verschijnsel hoeft niet alleen op te treden bij objecten uit verschillende niveaus. Bijvoorbeeld in bepaalde stadia kunnen plant en stengel samenvallen, zonder dat daarmee in die stadia de plant de stengel is. (Dit voorbeeld is afkomstig van S.A.Kripke.) Verder moeten we oppassen voor orgiën van reïficatie. Neem nu de postbode. 'Postbode' is een rol van een mens. Het gaat niet aan een speciaal object te postuleren: de postbode, met onderliggende mens.

[10] Ik hoop dat mijn gebruik van het woord 'gemedieerd' niet verwarrend is. De fysische processen zitten niet *tussen* de niet-fysische oorzaak en het niet-fysische gevolg in. Als ik bijvoorbeeld een boom zie, is er niet *éérst* de boom, *dan* allerlei tussenliggende fysische processen in ether, oog en brein en *dan* tenslotte de geheimzinnige gebeurtenis van het 'zien'. Op het niveau van het zien is er alleen maar dit: ik, de boom, mijn zien ervan. We kunnen spreken van directe waarneming. De onderliggende fysische processen maken deze waarneming mogelijk. Het idee van tussenliggende fysische processen —we zouden dat *de Mythe van het Medium* kunnen noemen— berust op een niveauverwarring.

Thesen (a) en (e) beschrijven de speciale status van het fysische niveau: het is het enige causaal zelfstandige. De hier beschreven versie van Fysicalisme is goed te verenigen met het Functionalisme, dat functionele niet-fysische toestanden postuleert. Dit Fysicalisme richt zich wel tegen de opvatting (à la parapsychologie) dat er een andere werkelijkheid zou zijn met een andere oorzakelijkheid.

2.3 Fysicalisme impliceert Mechanisme niet

Niets in de hierboven besproken liberale variant van het Fysicalisme sluit uit dat machines en organismen van heel verschillende aard zijn. Objecten van *geen van beide* soorten hoeven objecten op het fysisch niveau te zijn. Misschien hoort het wel essentieel bij machines dat ze een eindig aantal discrete functionele toestanden hebben, terwijl het begrip *toestand van een organisme* geen duidelijke betekenis heeft.

Maar als we een mens nu eens molecuul voor molecuul nabouwen? In de eerste plaats is dat zo'n ver verwijderde mogelijkheid dat we, denk ik, niet erg goed begrijpen wat 'molecuul voor molecuul' nabouwen nu eigenlijk betekent. In de tweede plaats is het niet echt duidelijk dat we het resultaat van dat bouwen zouden moeten beschrijven als 'machine'. Ten derde, wie weet krijgen we wel zowel een machine als een mens, zonder dat die twee identiek zijn.

2.4 Mechanisme impliceert Fysicalisme niet

Het lijkt me bijvoorbeeld dat het Functionalisme als theorie van het mentale niet getrouwd hoeft te zijn met één of andere versie van Fysicalisme. (Het Functionalisme ontleent natuurlijk wel veel van haar aantrekkelijkheid aan haar compatibiliteit met het Fysicalisme.) Het begrip 'functionele toestand' is nog steeds zinnig als er twee causaal zelfstandige niveaus van werkelijkheid zijn (aannemende dat zoiets überhaupt denkbaar is). Waarom zouden zowel organismen als machines niet aan beide niveaus deel kunnen hebben?

Lucas bestrijdt met zijn Argument het Mechanisme. Of Lucas' beweringen ook anti-fysicalistische implicaties hebben is niet zo duidelijk.

3 Machines, formele systemen, Gödelzinnen

Om duidelijk te maken waarvoor de uitleg in deze sectie nodig is, geef ik eerst een schematische versie van Lucas' Argument. Het is niet de bedoeling dat de lezer nu al de verschillende vaktermen die in het argument een rol spelen begrijpt.[11]

[11] Aan te bevelen literatuur: éénvoudige, vlot leesbare introducties zijn te vinden in: [C$^+$72], [Hof79], [NN58]. Uitstekende leerboeken zijn [BJ74] en [Dal04]. Het allermooiste —maar helaas een stuk moeilijker— is nog altijd het grandioze artikel [Fef60]. Voor historisch geïnteresseerden is het aan te bevelen [FDK$^+$86] te raadplegen. Voor Gödels biografie, zie [Daw97]. Een meer filosofisch getinte bespreking van Gödels leven en werk is [Wan87].

Lucas beweert dat wij mensen substantieel van machines verschillen. Er is iets waarin wij elke machine de baas zijn: het Gödeliseren. Het argument gaat zo:

i. Elke machine is een instantiatie (=belichaming) van een formeel systeem.

ii. Dus, gegeven een machine die consistent is en in staat is om eenvoudige rekenkunde te doen, is er —volgens de stellingen van Gödel— een ware rekenkundige zin die de machine niet zal produceren, namelijk de Gödelzin van het systeem dat hoort bij de machine.

iii. Wij kunnen echter de waarheid van die Gödelzin inzien.

iv. Ergo, machines vormen geen adequaat model van de menselijke geest.

De opzet van deze sectie is als volgt. Eerst leg ik uit wat we in deze context onder machines moeten verstaan (Subsectie 3.1) en wat onder formele systemen (Subsectie 3.2). En passant, sta ik stil bij de dubbele relatie tussen machines en formele systemen: een machine kan gezien als een instantiatie van een formeel systeem *en* een machine kan de stellingen van een formeel systeem volgens de regels van dat systeem produceren. Vervolgens bespreek ik een specifiek, in dit verband belangrijk, formeel systeem: de Rekenkunde (Subsectie 3.3). De Gödelstellingen worden in Subsectie 3.4 geïntroduceerd, vergezeld van een schets van hun bewijs. Subsectie 3.5 bevat een verrassing: ik geef een voorbeeld van een niet-formeel (maar wel precies beschreven) systeem. We zullen zien dat we ons heel goed een machine kunnen voorstellen die stellingen produceert volgens de regels van dit systeem. Dit niet-formele systeem zal een rol spelen bij onze bespreking van de stappen (ii) en (iii) van Lucas' Argument.

3.1 Machines

Bij *machine* in Lucas' Argument moeten we ons duidelijk niet een stoommachine of weefgetouw voorstellen, maar een geprogrammeerde digitale computer. De computer heeft een eindig aantal mogelijke discrete interne machinetoestanden en een, in elk stadium eindig, maar in principe willekeurig uitbreidbaar geheugen.[12] Bovendien is hij uitgerust met een invoerpoort en een uitvoerpoort. Via de invoerpoort kunnen we de momentane machinetoestand en/of de momentane geheugeninhoud wijzigen. Via de uitvoerpoort komt bepaalde geselecteerde informatie aangaande de interne toestand en de geheugeninhoud beschikbaar. Het rekenproces verloopt stapsgewijs. Een rekenstadium wordt volledig gegeven door de interne machinetoestand en de inhoud van het geheugen. Het programma specificeert een aantal regels om (we nemen hier aan: deterministisch) van rekenstadium naar rekenstadium te gaan. Bij het doen van zo'n rekenstap verandert de interne toestand van de machine en/of wordt de geheugeninhoud gewijzigd.

[12] In feite is wat we tot de interne toestanden rekenen en wat tot het geheugen een kwestie van conventie.

Als je aan een werkende computer denkt zijn er eigenlijk twee dingen: de abstracte machine —het ontwerp— en haar fysische realisatie. De fysische realisatie zal zich natuurlijk niet altijd conform het ontwerp gedragen: we spreken dan van een hardware-fout. We zullen zien dat de abstracte machine gerepresenteerd kan worden als een formeel systeem, de fysische machine is dan een instantiatie van dat systeem.

3.2 Formele systemen

Een formeel systeem wordt gegeven door drie dingen: zijn taal, zijn axioma's en zijn afleidingsregels.

3.2.1 De taal

De taal van het systeem wordt precies vastgelegd door een uitputtende beschrijving van de simpelste elementen van die taal (letters, haakjes, enzovoorts) en door instructies hoe je uit elementen van de taal die je al geconstrueerd hebt nieuwe kunt maken.

> Een enigszins gesimplificeerd voorbeeld. We beschrijven de taal van een formeel systeempje dat we MINI zullen dopen. De zinnen van de taal van MINI zijn als volgt gegeven:
>
> a. P zit in de taal.
>
> b. Als ϕ en ψ in de taal zitten, dan zit ook $(\phi \& \psi)$ erin.
>
> c. Niets zit in de taal dan wat op grond van (a) en (b) toegelaten is.
>
> In de taal zitten bijvoorbeeld: P, $(P\&P)$, $(P\&(P\&P))$, $((P\&P)\&P)$, $((P\&P)\&(P\&P))$; maar niet: $))PP\&PP(($ en ook niet: $(P\&Q)$.

3.2.2 De axioma's

De axioma's zijn een aantal (eventueel oneindig veel) zinnen uit de taal die als uitgangspunt dienen van formele bewijzen. We eisen dat de axioma's *effectief opsombaar* zijn. Dat betekent dat een digitale computer in principe een (eventueel oneindig doorgroeiende) lijst van axioma's moet kunnen produceren waarin elk axioma op den duur voorkomt.

3.2.3 De regels

De regels van een formeel systeem zijn instructies hoe je beginnende met de axioma's formele bewijzen kunt construeren. (We eisen net als bij de axioma's dat de regels effectief opsombaar zijn.) Een *formeel bewijs* is een eindig rijtje

zinnen: $\phi_1; \phi_2; \phi_3; \ldots; \phi_n$. Elk van deze zinnen is een zin uit de taal van het systeem. Een zin kan in het rijtje voorkomen op één van twee mogelijke gronden: de zin is een axioma of hij is door toepassing van een regel uit voorafgaande zinnen in het rijtje te concluderen. ϕ_n is de conclusie van het bewijs. Een zin is *bewijsbaar* in het systeem als er een bewijs is met die zin als conclusie. We noemen bewijsbare zinnen ook *stellingen*.

> MINI heeft als axioma: $((P\&P)\&P)$ en als regels: uit $(\phi\,\&\,\psi)$ mag je ϕ concluderen en uit $(\phi\,\&\,\psi)$ mag je ψ concluderen Een voorbeeld van een bewijs is nu:
>
> $$((P\&P)\&P); (P\&P); P$$
>
> MINI heeft precies vijf bewijzen en precies drie stellingen. Welke zijn dit? (MINI is natuurlijk maar een heel simpel voorbeeld. De formele systemen waarin je wiskunde kunt formalizeren hebben oneindig veel bewijzen.)

De begrippen 'formeel systeem' en 'formeel bewijs', zoals hier geïntroduceerd, zijn te beschouwen als puur syntactische noties. Syntaxis gaat over allerlei eigenschappen van talen die niet te maken hebben met betekenis. Tot het domein van de Syntaxis behoren operaties (bijvoorbeeld concatenatie) op talige entiteiten (zoals letters, woorden, zinnen).

3.2.4 Het formele systeem van een machine

Zoals we gezien hebben, gaat een machine stapsgewijs van rekenstadium naar rekenstadium. Een rekenstadium wordt gegeven door: één uit een eindig aantal interne toestanden, plus de geheugen-inhoud. Hierboven werd reeds beschreven dat de geheugentoestand in elk stadium eindig is. Een rekenstadium is dus per definitie eindig. We kunnen zulke rekenstadia representeren door zinnen van een formeel systeem. Op deze manier kunnen we de abstracte machine representeren als formeel systeem. Als de machine bijvoorbeeld drie interne toestanden: $S1$, $S2$, $S3$ heeft en de mogelijke geheugeninhouden bestaan uit rijen nullen en enen, dan zouden: $S1; 0010$, $S2; 01110$, en $S3; 111$ zulke zinnen kunnen zijn.

We kunnen de rekenregels van de machine vertalen naar afleidingsregels van ons systeem. Het beginstadium van het rekenproces laten we corresponderen met één uniek axioma. Bewijzen in ons systeem zullen overeenkomen met rijtjes van opeenvolgende rekenstadia vanaf de begintoestand. Mocht de machine stoppen, dan wordt de hele rekenprocedure beschreven door het langste bewijs in het systeem.

We hebben nu de machine abstract gerepresenteerd als formeel systeem, zodat we de fysische machine kunnen zien als instantiatie of belichaming van dit systeem.

Merk op dat de formele systemen die bij machines horen heel specifiek zijn: zo is er maar één axioma en kunnen bewijzen op hoogstens één manier met één zin verlengd worden.

3.2.5 Machines kunnen formele systemen opsommen

Behalve dat machines beschreven kunnen worden met behulp van een formeel systeem, kunnen ze ook worden gebruikt om zelf een formeel systeem te produceren. De stellingen van een formeel systeem kunnen worden opgesomd door een geschikt geprogrammeerde machine. Opsommen betekent hier dat een reeks stellingen uit de uitvoerpoort van de machine zal komen, waarbij elke stelling van het systeem ooit zal verschijnen. Ik zal zeggen dat de machine het formele systeem opsomt als zij niet alleen de stellingen van het systeem produceert, maar dit ook doet volgens de regels van het systeem; ik bedoel hiermee dat de machine intern de axioma's van het systeem genereert, bewijzen bouwt uit die axioma's volgens de regels van het systeem en tenslotte de conclusies van deze bewijzen als uitvoer presenteert. Het hoeft echter niet zo te zijn dat de regels van het opgesomde systeem identiek zijn aan de regels volgens welke de machine werkzaam is (de regels van het formele systeem dat de machine abstract beschrijft zoals in Subsubsectie 3.2.4 besproken). Het kan heel goed zijn dat de machine vele rekenstappen nodig heeft om één bewijsstap van het opgesomde systeem uit te voeren. Bovendien zal de machine de bewijzen van het formele systeem in een bepaalde volgorde moeten aflopen. De machine zal deze volgorde moeten bepalen. De instructies voor het bepalen van deze volgorde zullen in het algemeen geen deel uitmaken van het formele systeem dat door de machine wordt geproduceerd.

3.2.6 Interpretatie

De systemen waarvoor de Gödelstellingen geformuleerd zijn, zijn interpreteerbaar. We kunnen aan de symbolen van deze systemen betekenissen toekennen, zodat de zinnen van het systeem —gegeven deze betekenissen— opgevat kunnen worden als uitspraken. De verleende betekenissen maken de taal van het systeem pas echt tot taal in de gebruikelijke zin des woords: namelijk iets waarmee je iets kunt zeggen.

Bezie nu een systeem, waarvoor een interpretatie gespecificeerd is. We willen natuurlijk graag dat de axioma's waar zijn. (Een zin is waar als wat die zin zegt zo is; "Er staat een raket naast de Dom", bijvoorbeeld, is precies dan waar als er daadwerkelijk een raket naast de Dom staat. 'Waarheid' is ook van toepassing op de taal van ons systeem, nu zijn zinnen onder de gegeven interpretatie iets uitdrukken.) Verder is het gewenst dat de regels van het systeem correct zijn. Een regel is correct wanneer hij bij ware premissen altijd een ware conclusie oplevert. Omdat het soms erg moeilijk is om in te zien of axioma's waar zijn, behouden we ons het recht voor af en toe geïnterpreteerde systemen met eventueel onware axioma's te beschouwen.

We interpreteren de taal van MINI. Laat "P" staan voor *de Dom staat in Utrecht*. Laat "&" staan voor: en. Nu wordt "$(P\&P)$" geínterpreteerd als: *de Dom staat in Utrecht en de Dom staat in Utrecht*. Merk op dat de haakjes geen interpretatie krijgen: ze functioneren slechts als hulpsymbolen.

Ik geef toe: geen erg inspirerend voorbeeld. De lezer overtuige zichzelf van de waarheid van het axioma en de correctheid van de twee regels van MINI ten opzichte van de gegeven interpretatie. (MINI is *onvolledig* t.o.v. de gegeven interpretatie: er zijn ware stellingen die niet bewijsbaar zijn.)

3.2.7 Rijkdom

De formele systemen waarvoor de Gödelstellingen geformuleerd zijn, bevatten doorgaans de axioma's en regels van de zogeheten eerste orde predikatenlogica, plus de axioma's van de Elementaire Rekenkunde. We zullen zulke systemen rijk noemen. De eerste orde predikatenlogica codificeert het gewone wiskundige redeneren: zij bevat de regels voor het gebruik van uitdrukkingen als: en, of, niet, als... dan, voor alle, er is een. Systemen in de eerste orde predikatenlogica hebben oneindig veel bewijzen; de totaliteit van deze bewijzen is in het algemeen zeer gecompliceerd. De Rekenkunde wordt behandeld in Sectie 3.3.

3.2.8 Consistentie

Een formeel systeem heet consistent als niet elke zin uit de taal van het systeem bewezen kan worden. Als een systeem de eerste orde predikaten logica omvat kunnen we ook een andere definitie geven: het systeem is consistent als er geen zin ϕ is, zodanig dat zowel ϕ als de ontkenning van ϕ bewezen kunnen worden. Of, nog anders: als we geen tegenspraak van de vorm 'ϕ en niet-ϕ', kunnen bewijzen. Hoe kunnen we erachter komen of een gegeven systeem consistent is? Helaas, er is geen algemene procedure om de consistentie te bepalen. Bezie maar eens een formeel systeem met oneindig veel bewijzen; consistentie voor zo'n systeem betekent dat er een zin ϕ is zodat er geen bewijs is van ϕ. We moeten dus inzien dat geen bewijs ϕ als conclusie heeft. De naïeve manier om hier achter te komen zou zijn de bewijzen één voor één te bekijken en te controleren dat hun conclusie anders dan ϕ is. Omdat —naar we hebben aangenomen— er oneindig veel bewijzen zijn komt deze procedure, als ϕ inderdaad niet als conclusie voorkomt, niet tot een eind. Het begint er nu naar uit te zien dat we er nooit achter kunnen komen of een formeel systeem met oneindig veel bewijzen consistent is. Gelukkig kunnen we in veel gevallen andere procedures bedenken

dan de naïeve. Stel, bijvoorbeeld, je hebt een geïnterpreteerd systeem in de eerste orde predikatenlogica. Stel ook dat we kunnen inzien dat de axioma's waar zijn en de regels correct. Dan is het niet moeilijk aan te tonen dat alle stellingen van het systeem waar zijn. Er volgt nu dat het systeem consistent is. Als zowel ϕ als de negatie van ϕ bewijsbaar waren, zouden immers zowel ϕ als de negatie van ϕ waar zijn. Dit laatste is onmogelijk.

3.2.9 Formele bewijzen en echte bewijzen

Er is iets vreemds aan de hand met de notie 'bewijs' sinds Euclides.[13] De begrippen 'axiomatisch bewijs' en, later, 'formeel bewijs' waren zo helder en zo succesvol dat men het alledaagse begrip als vaag, twijfelachtig en zelfs non-existent ging beschouwen.[14] "Echte bewijzen vind je alleen in Wiskunde en Logica." Ik deel dit scepticisme niet Ik geloof dat echte bewijzen even gewoon zijn en even veelvuldig voorkomen als fietsen, tafels, bomen enzovoorts. Het lijkt me dan ook zinnig je af te vragen of formele bewijzen bewijzen zijn.

Het is zeker niet zo dat alle formele bewijzen ook echt bewijzen zijn: bijvoorbeeld een bewijs in het formele systeem van een machine is zeker geen bewijs in enige gebruikelijke zin. We kunnen immers de 'zinnen' van het systeem niet zien als uitspraken in enige gebruikelijke zin. Het lijkt me echter dat *als* we een geïnterpreteerd systeem in gedachten hebben en *als* we van de axioma's van dat systeem inzien dat ze waar zijn en *als* we van de regels inzien dat ze correct zijn dat we dan zeker kunnen zeggen dat de bewijzen van dat systeem ook echte bewijzen zijn (of misschien beter: realiseren) t.o.v. de gegeven interpretatie. Je zou in dit geval kunnen zeggen:

> (formeel bewijs + interpretatie + inzicht in waarheid van axioma's en correctheid van regels voor gegegeven interpretatie) = echt bewijs.

Zelfs als we een interpretatie gegeven hebben zijn de de begrippen *formeel bewijs* en *echt bewijs* duidelijk verschillend. Het is in het gewone begrip *bewijs* ingebouwd dat *als* je een bewijs van iets hebt dat *dat* dan ook waar is. Bijvoorbeeld als ik een bewijs heb dat sneeuw wit is, dan is sneeuw ook wit. Maar geeft het hebben van een bewijs op die manier dan absolute zekerheid? Nee, we kunnen ons namelijk vergissen in die zin dat we abusievelijk denken dat we een bewijs hebben, terwijl datgene wat we voor een bewijs houden er geen is. Ik kan bijvoorbeeld denken dat ik een bewijs heb dat $733 \times 721 = 508493$ —namelijk een berekening volgens het gewone algorithme voor decimale vermenigvuldiging—, terwijl ik in werkelijkheid een rekenfout gemaakt heb. In dat geval heb ik geen

[13] Deze sectie bevat een paar van mijn eigen ideeën over echte bewijzen. De meeste logici zullen mijn opinies over deze materie waarschijnlijk niet delen.

[14] Zo zegt Lucas in [Luc68], p174: "Many people do speak colloquially of their being able to prove things a machine cannot prove, and it would be permissible to define an absolute provability to accomodate this locution. Permissible, but inexpedient. Provability has been construed by mathematical logicians for a generation as a syntactical term with a very precise definition, and it could be confusing to import loose notions of what I can see to be true into this well-disciplined and useful concept."

berekening volgens het gewone algorithme uitgevoerd en heb ik ook geen bewijs. Bezie nu eens een geïnterpreteerd formeel systeem, zeg S, en stel dat we de waarheid van de axioma's van S en de correctheid van de regels van S inzien. Formele bewijzen in S zullen nu ook echte bewijzen zijn (realiseren). Laten we nu ook nog aannemen dat elk bewijs dat we überhaupt in de taal van S kunnen formuleren correspondeert met een formeel bewijs van S. Ik beweer dat zelfs in dit geval de begrippen *formeel bewijs in S* en *bewijs in de taal van S* niet samenvallen. Mijn argument is dit. Zelfs al zien we terecht in dat de axioma's waar zijn en de regels correct, dan is er toch altijd de mogelijkheid dat we ons vergissen. Als we ons vergissen, *dan* zijn de formele bewijzen van S nog steeds formele bewijzen van S, ze zijn echter geen bewijzen meer. We zien dus dat de formele bewijzen van S bewijzen zijn, maar dat het nog steeds voorstelbaar is dat ze het niet zijn. Ergo, de begrippen *formeel bewijs in S* en *bewijs in de taal van S* vallen niet samen.

Het hele eier-eten is dat bewijsbaarheid *een begrip is in termen waarvan we cognitief succes formuleren, terwijl een formeel systeem je een proces geeft waarmee je eventueel bewijzen kunt produceren.* Vergelijk het met een stenenfabriek: aan de ene kant heb je de criteria waaraan een goede steen moet voldoen, aan de andere kant heb je een productieproces dat zulke goede stenen oplevert. Ook al levert het proces de facto goede stenen, nog steeds zijn 'goede steen' en 'steen geproduceerd volgens het proces' niet synoniem: we kunnen ons voorstellen dat om een of andere reden die we totaal over het hoofd hebben gezien het proces plotseling slechte stenen zou gaan opleveren. *Goede steen* correspondeert hier met *bewijs*, *proces* met *formeel systeem*.

3.3 De Rekenkunde

De Rekenkunde (meestal *Peano Rekenkunde* genoemd) is een formeel systeem in de eerste orde predikatenlogica met een vaste interpretatie. De Rekenkunde gaat over de (natuurlijke) getallen: $0, 1, 2, \ldots$ Je kunt in de taal zulke dingen uitdrukken als:

- $7 + 5 = 12$;
- $7 \cdot 5 = 35$;
- voor alle getallen m en n hebben we $m + n = n + m$ (commutativiteit van de optelling);
- voor alle getallen n is er een priemgetal p, met $p > n$.[15] (M.a.w. er zijn oneindig veel priemgetallen.)
- voor alle getallen m en n hebben we $m + (n + 1) = (m + n) + 1$.

In de Rekenkunde kun je bovenstaande zinnen ook bewijzen. De laatste zin is een voorbeeld van een axioma. Het is niet moeilijk in te zien dat de (hier

[15] Een priemgetal is een natuurlijk getal dat groter is dan 1 en dat alleen deelbaar is door 1 en door zichzelf.

niet gegeven) axioma's van de Rekenkunde waar zijn en de regels correct. Er volgt dat je in de Rekenkunde alleen maar ware uitspraken over getallen kunt bewijzen en dat de Rekenkunde consistent is.

Bekijk nu eens de volgende zin: voor alle getallen n zijn er priemgetallen p en q zodat $p - q = 2$ en $q > n$ (het zogeheten Priemtweelingen Vermoeden). We hebben niet het flauwste idee of deze zin waar is; we hebben ook geen idee of deze zin —aannemende dat hij waar is— bewijsbaar is. De enige manier die we tot nu toe hebben kunnen bedenken om erachter te komen of die zin bewijsbaar is, is de naïeve procedure van het één voor één aflopen van alle bewijzen, en die procedure hoeft niet tot een eind te komen.

Kan het überhaupt voorkomen dat een rekenkundige zin waar is, en niet bewijsbaar in de Rekenkunde? Als je een willekeurig voorbeeld van een rekenkundige zin bekijkt waarvan je *weet* dat hij waar is —zoals: elk getal kan op precies één manier ontbonden worden in priemfactoren— dan blijkt deze ook in de Rekenkunde bewijsbaar zijn. De eerste Gödelstelling vertelt ons dat deze empirie bedrieglijk is: er is een ware rekenkundige zin die niet in de Rekenkunde bewijsbaar is.

3.4 Gödels Onvolledigheidsstellingen

Er zijn twee onvolledigheidsstellingen. We concentreren ons voorlopig op de eerste. Gödels Eerste Onvolledigheidsstelling vertelt ons, het volgende. Voor elk consistent formeel systeem dat voldoende Rekenkunde omvat, m.a.w. dat rijk is (zie Subsubsectie 3.2.7), is er een ware zin —de Gödelzin van dat systeem— die niet bewijsbaar is in dat systeem. (Waarheid is hier: waarheid onder de gewone interpretatie van de rekenkunde.) Bovendien is onder zekere verdere condities de negatie van de Gödelzin ook niet bewijsbaar.[16]

Het bewijs van de Eerste Onvolledigheidsstelling is geïnspireerd door een paradox over echte bewijsbaarheid: de bewijsbaarheidsparadox. Deze paradox is nauw verwant aan de beroemde Leugenaarsparadox van Epimenides. Bekijk de volgende zin: *deze zin is niet bewijsbaar.* Veronderstel dat die zin bewijsbaar is. Dan is hij waar, immers alleen ware dingen zijn bewijsbaar. Maar die zin beweert van zichzelf dat hij niet bewijsbaar is. Dus als hij waar is, dan is hij onbewijsbaar. Er volgt —nog steeds aannemende dat die zin bewijsbaar is— dat de zin niet bewijsbaar is. Een tegenspraak. Het uitgangspunt dat de zin bewijsbaar is, leidt tot een tegenspraak en kan dus niet waar zijn. We moeten concluderen dat die zin niet bewijsbaar is. Maar dat laatste is precies wat de zin zegt en daarmee hebben we de zin bewezen! We hebben dus een onbewijsbare zin bewezen: een *paradox*!

Wat er mis is met bovenstaande redenering is moeilijk te zeggen. Filosofen zitten elkaar daarover nog steeds in de haren.[17] Over die vraag wil ik het hier

[16] Er bestaat een andere zin, de zogenaamde Rosserzin van het systeem, waarvan je kunt laten zien dat *als* het systeem dat we beschouwen rijk genoeg is en consistent, dat noch die zin, noch zijn negatie bewijsbaar zijn in het systeem. Er zijn nog veel wildere zinnen uitgevonden; deze vallen helaas buiten het bestek van deze voetnoot ...

[17] Wat mij buiten kijf lijkt te staan is dat deze paradox een *semantische paradox* is. Wat er

niet hebben: het gaat immers om de Gödelstelling. Het idee van het bewijs van de Eerste Gödelstelling is als volgt: we gaan bovenstaande paradox zo goed mogelijk nabouwen, maar dan voor formele bewijsbaarheid in systemen die voldoende Rekenkunde bevatten. Omdat formele bewijsbaarheid een syntactisch, en dus glashelder begrip is, zullen we niet echt tot een paradox komen: de tot een paradox leidende redenering zal op subtiele manier geblokkeerd blijken te zijn. De beste benadering van de tot een paradox leidende redenering geeft ons echter een fraaie bonus: de Eerste Onvolledigheidsstelling. Om de gedachten te bepalen nemen we als formeel systeem de Rekenkunde zelf.

Laten we het eerst eens naïef proberen en leren van wat er niet gaat. Onze eerste poging om de paradoxale zin te formuleren wordt: *deze zin is niet bewijsbaar in de Rekenkunde*. Het zij duidelijk dat deze zin niets interessants kan opleveren: hij is op flauwe manier waar. Het is namelijk helemaal geen zin uit de taal van de Rekenkunde en dus zeker niet bewijsbaar in de Rekenkunde. Er zijn in feite twee niet-rekenkundige ingrediënten: '*formele bewijsbaarheid* is geen rekenkundig begrip maar een begrip uit de elementaire Syntaxis *en* het indexicale woord "deze" komt niet voor in de rekenkundige taal. *De grote vondst van Gödel is het idee om bovenstaande zin te 'vertalen' in de taal van de Rekenkunde door de elementaire Syntaxis in die taal te vertalen.*

Laat ik eerst aanduiden hoe we het gebruik van "deze" kunnen vervangen door een toepassing van een elementair syntactische operatie.

3.4.1 De eliminatie van 'deze'

De truc die we hier presenteren is afkomstig van W.V.O. Quine. Bezie bijvoorbeeld: *deze zin is leuk*. Quines parafrase is als volgt:

- ▷ *"geeft iets leuks wanneer je het achter zijn eigen aanhaling zet." geeft iets leuks wanneer je het achter zijn eigen aanhaling zet.*

De *aanhaling* van een aantal woorden is wat je krijgt als je die woorden tussen aanhalingstekens zet. Wat moeten we volgens onze zin tussen aanhalingstekens zetten? De woorden die in die zin tussen aanhalingstekens gepresenteerd worden! Het resultaat van deze operatie is de zin zelf. De zin zegt dus van zichzelf dat hij leuk is. Precies het resultaat dat we wilden hebben. De beschreven operatie: een aantal symbolen achter hun eigen aanhaling zetten, is van elementair syntactische aard.[18]

3.4.2 Van elementaire Syntaxis naar Rekenkunde

Om de vertaling van elementaire Syntaxis naar Rekenkunde uit te leggen, beperk ik me tot een zeer klein deeltheorietje van de elementaire Syntaxis.[19] Het is

achter deze paradox zit, moet het zelfde zijn als wat er achter, bijvoorbeeld, de Leugenaarsparadox zit.

[18] Je kunt een analoog idee gebruiken om een elementair computerprogramma te schrijven dat zichzelf uitprint.

[19] De hier gevolgde benadering gaat terug op [Smu61].

de theorie van reeksen van de letters `A` en `B` met de operatie concatenatie (= achter elkaar zetten). Voorbeelden van zulke reeksen zijn `ABBA` en `AABAAB`. De concatenatie van deze reeksen is: `ABBAAABAAB`. We schrijven: `ABBA` $\star$ `AABAAB` = `ABBAAABAAB`. In de theorie hebben ook nog $\Box$, de lege tekenreeks, en variabelen x, y, z voor willekeurige tekenreeksen.

> Het symbool '$\Box$' is een *naam* voor de lege tekenreeks; het is natuurlijk niet *zelf* de lege tekenreeks. Dit is wat verwarrend omdat bijvoorbeeld '`ABBA`' niet slechts fungeert als *naam* van de tekenreeks `ABBA`, maar ook als *voorbeeld* van die tekenreeks zelf!
>
> Enige kenmerken van concatenatie zijn:
>
> - $x \star \Box = \Box \star x = x$.
> - $(x \star y) \star z = x \star (y \star z)$.
> - Als $x \star y = x \star z$, dan $y = z$.

We gaan nu ons theorietje vertalen naar de Rekenkunde. Dit doen we door aan `A`, `B`, $\Box$ getallen toe te kennen en concatenatie te representeren door een rekenkundige operatie $\circledast$. We gebruiken een hulpfunctie ℓ. We nemen:

- $\Box \mapsto 0$;
- `A` $\mapsto 1$;
- `B` $\mapsto 2$;
- $\ell(n) :=$ de grootste macht van 2 kleiner of gelijk aan $(n+1)$);
  (bijvoorbeeld, $\ell(0) = 1$, $\ell(1) = 2$, $\ell(2) = 2$, $\ell(3) = 4$;)
- $m \circledast n := m \cdot \ell(n) + n$;
  (bijvoorbeeld, $7 \circledast 5 = 7 \cdot \ell(5) + 5 = 7 \cdot 4 + 5 = 33$.)[20]

We representeren nu concatenatie $\star$ door de rekenkundige operatie $\circledast$.

> Wil $\circledast$ de operatie concatenatie geschikt representeren dan moet $\circledast$ de verschillende eigenschappen van concatenatie weerspiegelen en moet dit feit in de Rekenkunde verifieerbaar zijn. Bijvoorbeeld: $7 \circledast 0 = 7 \cdot 1 + 0 = 7$, en $(3 \circledast 4) \circledast 5 = (3 \cdot 4 + 4) \cdot 4 + 5 = 69 = 3 \cdot 16 + (4 \cdot 4 + 5) = 3 \circledast (4 \circledast 5)$.

[20] Het is een elementaire oefening is om de definitie van $\circledast$ te motiveren. Daarom is het opmerkelijk hoe makkelijk het is om de definitie te verprutsen. Dit gebeurt vaak in de literatuur. Eén betreurenswaardig voorbeeld is de behandeling in [Vis86].

We vertalen nu ABBA, dat is A ⋆ B ⋆ B ⋆ A, als $1 \circledast 2 \circledast 2 \circledast 1$, dat is: 21.[21]

> Vertaling zoals we dat hier ingevoerd hebben heet in wiskundige taal: *inbedding in een definitionele uitbreiding*. Een belangrijk kenmerk hiervan wordt gegeven door het volgende plaatje:
>
> $$\begin{array}{ccccc} \texttt{AB} & \star & \texttt{BA} & = & \texttt{ABBA} \\ \downarrow & \downarrow & \downarrow & & \downarrow \\ 4 & \circledast & 5 & = & 21 \end{array}$$

3.4.3 Het bewijs

We hebben nu de ingrediënten bijeen voor het bewijs van de Eerste Onvolledigheidsstelling. We hebben laten zien hoe we het indexicale "deze" kunnen elimineren en hoe we de elementaire Syntaxis kunnen vertalen naar de Rekenkunde. We kunnen nu een rekenkundige zin construeren die hetzelfde effect heeft als *deze zin is niet bewijsbaar in de Rekenkunde*. Laten we deze rekenkundige zin "G" dopen. Deze zin is natuurlijk niet paradoxaal: het zou toch te gek voor woorden zijn als een puur rekenkundige zin paradoxaal was.[22]

We redeneren nu als volgt.

> Stel G is formeel bewijsbaar in de Rekenkunde, dan is G ook waar. Immers, de axioma's van de Rekenkunde zijn waar en de regels zijn correct. Maar als G waar is, dan is G niet formeel bewijsbaar in de Rekenkunde. Dus, gesteld dat G formeel bewijsbaar is in de Rekenkunde, volgt dat G niet formeel bewijsbaar is in de Rekenkunde. Een tegenspraak. Er volgt dat G niet formeel bewijsbaar kan zijn in de Rekenkunde. Dit is echter weer wat G zegt. Dus G is ook nog waar.

Kunnen we niet verder gaan en een paradox afleiden door bovenstaande redenering om te zetten in een redenering in de Rekenkunde en daarmee G formeel te bewijzen in de Rekenkunde? Nee dat gaat niet: de stap *als G formeel bewijsbaar is in de Rekenkunde, dan is G waar* is niet verifiëerbaar in de Rekenkunde.

[21] De specifieke Gödelnummering die we hier uitgelegd hebben correspondeert precies met de alfabetische volgorde van woorden in puzzlewoordenboeken: we hebben:

0	□	5	BA	10	ABB	15	AAAA
1	A	6	BB	11	BAA	16	AAAB
2	B	7	AAA	12	BAB	17	AABA
3	AA	8	AAB	13	BBA	18	AABB
4	AB	9	ABA	14	BBB	19	ABAA

[22] De pointe is hier, volgens mij, dat *echte bewijsbaarheid*, net als *waarheid*, een *semantisch begrip* is. Van semantische begrippen zijn we paradoxen gewend. *Formele bewijsbaarheid* is een *syntactisch begrip*: dat kan geen aanleiding geven tot paradoxen.

We hebben gezien dat G waar is en niet bewijsbaar in de Rekenkunde. De negatie van G is onwaar en daarom ook niet bewijsbaar in de Rekenkunde. Hiermee hebben we de Eerste Onvolledigheidsstelling van Gödel bewezen voor het geval van de Rekenkunde.[23] Je kunt laten zien dat de stelling ook geldt voor iedere consistente, rijke theorie (zie Subsubsectie 3.2.7). De eis van 'rijkdom' (of een soortgelijke eis) is nodig om de vertaling van Syntaxis in de beschouwde theorie mogelijk te maken.

> Bekijk een ware theorie T die rijk genoeg is. We hebben gezien dat T niet de Gödelzin G van T bewijst. We weten bovendien dat de zin G waar is. Daarmee is $T+G$ weer een ware theorie. Maar hoe zit het nu met $T+G$ daarvoor moet de Gödelstelling toch ook gelden? Zeker, maar de Gödelzin voor $T+G$ is uiteraard een andere dan die voor T.
>
> Laten we de afhankelijkheid van de theorie van Gödelzinnen zichtbaar maken door G_T te schrijven voor de Gödelzin van T. De Gödelstelling laat zien dat, als we starten met een ware en voldoende rijke theorie T, we een rij steeds sterkere ware theorieën T, $T+G_T$, $T+G_T+G_{T+G_T}$, ... kunnen construeren.[a]
>
> [a] De lezer kan over dit soort hiërarchieën het prachtige artikel [Fef62] raadplegen. Helaas is het artikel alleen geschikt voor diegenen met enige achtergrond in de logica.

We hebben gezien dat machines formele systemen kunnen opsommen. In feite kun je de Gödelzinnen van opgesomde systemen op volledig mechanische manier construeren uit programma's van de opsommende machines, mits zulke programma's gegeven zijn in een vast gekozen programmeertaal.

In Subsubsectie 3.2.8 hebben we het begrip *consistentie* ingevoerd. *Consistentie* is een syntactische notie. We kunnen het consistentiebegrip dus vertalen in de taal van de Rekenkunde. De Tweede Onvolledigheidsstelling vertelt ons het volgende. Als een consistente theorie T rijk is, dan bewijst T de zin die de vertaling is in rekenkundige taal van T *is consistent* niet. In slogan (maar minder precies): een voldoende rijke theorie kan haar eigen consistentie niet bewijzen, tenzij die theorie inconsistent is. Het bewijs van de Tweede Onvolledigheidsstelling is een verfijning van het bewijs van de Eerste. Het voert te ver om het hier te reproduceren.

[23] In feite hebben we bij ons bewijs gebruik gemaakt van tamelijk zwaar geschut: we hebben het principe gebruikt dat wat bewijsbaar is in de Rekenkunde waar is. Het is mogelijk het bewijs te verfijnen zodat de enige substantiële onderstelling die je nodig hebt de consistentie van de Rekenkunde is.

Bekijk een theorie T die rijk genoeg is. Het gekke is dat *als* T bewijst dat zij consistent is, zij ipso facto inconsistent moet zijn. Het is daarentegen heel goed mogelijk dat een consistente theorie bewijst dat zij inconsistent is.

3.5 De Feferman Rekenkunde

Het opsommen van bewijzen in een formeel systeem vormt om vele redenenen geen goed model van het menselijk redeneren. Eén punt is dat wanneer een formeel systeem in de eerste orde predikatenlogica op een tegenspraak stuit, het hele systeem 'explodeert' en alles bewijsbaar wordt. Wij mensen concluderen wanneer we op een tegenspraak stuiten natuurlijk niet dat alles waar is, maar we onderzoeken de relevante hypothesen en verwerpen een paar van de meest verdachte. We kunnen heel goed mechanische procedures bedenken voor het verwerpen van axioma's. Eén zo'n procedure leidt tot de *Feferman Rekenkunde.*[24]

Bezie de gewone Rekenkunde (= Peano Rekenkunde). We verkrijgen nu de Feferman Rekenkunde als volgt. We beginnen eerst alle bewijzen van Peano Rekenkunde op een rij te zetten: $\pi_1, \pi_2, \pi_3, \pi_4, \ldots$ Zodra we een bewijs van een tegenspraak tegenkomen, kijken we wat grootste lengte L is van een rekenkundig axioma ϕ is in dat bewijs. We gooien nu alle bewijzen weg die gebruik maken van axioma's met lengte groter of gelijk aan L. Daarna gaan we door met opsommen, waarbij we alleen bewijzen accepteren die gebruik maken van rekenkundige axioma's korter dan L. Als we eventueel weer een bewijs van een tegenspraak tegenkomen, zoeken we weer het de grootste lengte van een rekenkundig axioma in dat bewijs en herhalen de procedure. Een zin heet nu *bewijsbaar in Feferman Rekenkunde* als er bij de hierboven beschreven procedure *ooit* een bewijs van die zin wordt geproduceerd zodanig dat dit bewijs *nooit* wordt weggegooid.

Feferman Rekenkunde is niet een formeel systeem in de gebruikelijke zin. Toch zou het misleidend zijn het een 'informeel' systeem te noemen: 'informeel' suggereert een gebrek aan precisie bij de beschrijving van het systeem; de bovenstaande beschrijving laat echter niets te wensen over.

De procedure van de Feferman Rekenkunde lijkt mij volstrekt redelijk. Het modelleert het gedrag van iemand die de axioma's en regels van de Peano Rekenkunde plausibel vindt, maar toch nog een slag om de arm houdt. Natuurlijk is de ingebouwde 'back tracking' procedure niet die welke een mens echt zou volgen: het verwerpen van axioma's gebeurt niet op grond van inzicht in hun

[24]De Feferman Rekenkunde is uitgevonden door Feferman. (Dit is geen tautologie: de Peano Rekenkunde werd uitgevonden door Dedekind.) Zie [Fef60]. De Feferman Rekenkunde is niet alleen filosofisch interessant: in de handen van Feferman en Orey werd zij een machtig hulpmiddel bij de wiskundige bestudering van Relatieve Interpreteerbaarheid. Mijn gebruik van de uitdrukking 'Feferman Rekenkunde' is idiosyncratisch: in de literatuur gaat het steeds over 'Fefermans Predicate'.

onjuistheid, maar op grond van een puur formeel criterium. Toch is de Feferman Rekenkunde al een opmerkelijk nette theorie. Het is bijvoorbeeld eenvoudig in te zien dat zij een theorie is in de eerste orde predikatenlogica.

Wij, die weten dat de Rekenkunde consistent is, kunnen inzien dat een rekenkundige zin bewijsbaar is in Fefermans systeem precies dan als hij bewijsbaar is in de gewone Rekenkunde. We zullen immers nooit een bewijs van een tegenspraak tegenkomen bij het opsommen van de bewijzen van de Rekenkunde, dus zal de in Fefermans systeem ingebouwde 'back tracking' procedure nooit worden geactiveerd. Toch zijn Fefermans systeem en dat van Peano naar intentie verschillend.

We kunnen het begrip *bewijsbaarheid in Feferman Rekenkunde* vertalen in de taal van de Rekenkunde. Dit stelt ons in staat het bewijs van Gödels Eerste Onvolledigheidsstelling te herhalen voor de Feferman Rekenkunde.[25] Wat we nodig hebben om de waarheid, en daarmee de niet bewijsbaarheid, van de Gödelzin van Fefermans systeem te bewijzen is niet de consistentie van Feferman Rekenkunde —dat is een te zwakke bewering: de consistentie is immers 'ingebouwd'—, maar de consistentie van Peano Rekenkunde.

De Tweede Onvolledigheidsstelling geldt niet voor Fefermans systeem. Wij kunnen zelf eenvoudig inzien dat Fefermans systeem consistent is, zelfs als we ons agnostisch opstellen ten aanzien van de consistentie van de Peano Rekenkunde. De eliminatie van tegenspraken is immers in het systeem ingebouwd! Het blijkt dat de redenering die aan dit inzicht ten grondslag ligt in de Rekenkunde (en dus ook in de Feferman Rekenkunde) vertaald kan worden. Dus bewijst de Feferman Rekenkunde haar eigen consistentie.

Het is duidelijk dat Fefermans systeem door een machine 'geproduceerd' kan worden. De machine volgt gewoon de hierboven beschreven procedure. Laten we ons de uitvoerpoort van de machine voorstellen als een scherm, waarop de voorlopig bewezen stellingen één voor één getoond worden. Mocht de machine op een tegenspraak stuiten, dan verschijnt "Ik herroep de volgende stellingen:" gevolgd door een lijst van de conclusies van de weggegooide bewijzen op het scherm. Daarna vertelt de machine ons: "Ik ga door met voorlopige stellingen" en komen er weer voorlopige stellingen op het scherm. Als we onwetendheid pretenderen aangaande de consistentie van de gewone Rekenkunde, is het echter niet duidelijk welke stellingen die op het scherm verschijnen blijvend zijn en dus tot de uiteindelijke stellingen van het systeem behoren.[26] Naast de machine

[25] De Eerste Onvolledigheidsstelling voor Fefermans systeem volgt ook direct uit de Eerste Onvolledigheidsstelling voor de gewone Rekenkunde, gecombineerd met het inzicht dat de Feferman Rekenkunde en de Peano Rekenkunde de zelfde stellingen bewijzen. De Gödelzin van Fefermans systeem verschilt echter op interessante manier van die van Peano's systeem.

[26] Wat kan de Feferman Rekenkunde over bewijsbaarheid in zichzelf zeggen? De situatie is nogal subtiel: voor elke rekenkundige zin ϕ geldt: als de Feferman Rekenkunde ϕ bewijst, dan bewijst de Feferman Rekenkunde ook dat de Feferman Rekenkunde ϕ bewijst. Dit betekent als het ware dat als de Feferman Rekenkunde iets bewijst, zij dit feit ook zelf kan inzien. Bezie echter de Gödelzin G van Fefermans systeem. Feferman Rekenkunde bewijst niet: *als* Feferman Rekenkunde G bewijst, dan bewijst Feferman Rekenkunde dat Feferman Rekenkunde G bewijst. Dit betekent dat de Feferman Rekenkunde niet zelf kan inzien dat *als* hij iets bewijst, dat hij dan ook bewijst dat hij dat iets bewijst.

die Fefermans systeem opsomt zou een machine kunnen staan die de gewone (Peano) Rekenkunde opsomt. We kunnen ons voorstellen dat de twee machines er van buiten precies hetzelfde uitzien en dat gelijktijdig dezelfde stellingen op beide schermen verschijnen. Omdat de Rekenkunde in feite consistent is zijn beide machines dus vanuit behaviouristisch oogpunt hetzelfde. In werkelijkheid volgen ze echter verschillende procedures!

4 Machina abscondita

We zullen de schematische versie van Lucas' Argument nog eens ten tonele voeren, om deze vervolgens aan te vullen en te bespreken.[27]

i. Elke machine is een instantiatie (=belichaming) van een formeel systeem.

ii. Dus, gegeven een machine die consistent is en in staat is om eenvoudige rekenkunde te doen, is er —volgens de stellingen van Gödel— een ware rekenkundige zin die de machine niet zal produceren, namelijk de Gödelzin van het systeem dat hoort bij de machine.

iii. Wij kunnen echter de waarheid van die Gödelzin inzien.

iv. Ergo, machines vormen geen adequaat model van de menselijke geest.

Hoe mooi zou het niet zijn als dit argument correct was. We zouden dan een eenvoudige weerlegging van het Mechanisme hebben die noch berust op een taalkundige goocheltruc[28] of op de metafysica van het lekkere gevoel in je maag (hoewel het belang van dat lekkere gevoel niet onderschat mag worden). Helaas ...

4.1 Ad (i)

De eerste stap is: elke machine is een instantiatie (=belichaming) van een formeel systeem. Machines zijn hier blijkbaar geprogrammeerde digitale computers. De bewering uit (i) heb ik al uitvoerig besproken in Subsectie 3.1 en de Subsubsecties 3.2.1 tot en met 3.2.4.

4.2 Ad (ii)

De tweede stap begint met "Dus, gegeven ...". Dat "gegeven" is een belangrijk punt in de discussie. Ik wil de bespreking ervan echter liever uitstellen tot we stap (iii) behandelen.

Verder is er sprake van een machine die consistent is en in staat is elementaire rekenkunde te doen. Maar wat betekent dat? Wat 'doet' de machine als hij rekenkunde doet? Deze vraag is van belang omdat de wedstrijd tussen mens

[27] Literatuur o.a. [Ben67], [Bow82], [Chi72], [Hof79], [Luc61], [Luc68], [Web83].

[28] Zoals: een robot kan geen gevoelens hebben, want per definitie is 'gevoelens hebben' alleen van toepassing op organismen, en robots zijn geen organismen.

en machine eerlijk moet zijn. Eerlijkheid brengt mee dat we hetzelfde soort handeling van beide contestanten vergelijken.[29]

Misschien doet de machine wel dit: hij produceert een groeiende lijst van rekenkundige beweringen. Deze beweringen moeten consistent zijn (in een nader te beschrijven zin) en de stellingen van de Rekenkunde omvatten. We moeten elke bewering zien als een definief geponeerde stelling die niet tijdens het verdere rekenverloop kan worden ingetrokken. Goed, maar wat *maakt* dat de machine 'definitieve beweringen doet' en niet bijvoorbeeld zomaar wat rekenkundige zinnen opsomt, of bezig is rekenkundige zinnen te ontkennen, of een 'back tracking' procedure zoals bij Fefermans systeem uitvoert, of grappen vertelt? Het antwoord moet waarschijnlijk luiden: de bedoelingen van de programmeur of de specificaties van het programma. Of is het misschien voldoende dat wij —als het ware ad hoc— de activiteit van de machine interpreteren als het poneren van stellingen?

Deze laatste manier van zien, vind ik niet erg plausibel. Je abstraheert dan volledig van het interne mechanisme, terwijl we kunnen inzien, als we bijvoorbeeld een machine die de stellingen van de Peano Rekenkunde opsomt door de bewijzen af te lopen vergelijken met een machine die de backtracking procedure van het Feferman systeem implementeert, dat alleen maar de output niet vastlegt wat er echt gebeurt.[30] Ik ben daarom geneigd te denken dat de bedoelingen-van-de-programmeur/programmaspecificaties theorie het meest in de goede richting zit. (Voor deze theorie is er wel het probleem dat een programma lang niet altijd doet wat de programmeur wil.)

Is het voldoende aan te nemen dat de machine definitieve rekenkundige stellingen opsomt? De formulering "in staat is eenvoudige rekenkunde te doen" suggereert dat de manier waarop de machine tot haar beweringen komt lijkt op het produceren van rekenkundige bewijzen. Aan de andere kant, zou die veronderstelling —denk ik— de toelaatbare programma's wel erg beperken. Is het wel zo duidelijk dat de manier waarop een menselijke wiskundige tot een rekenkundige stelling komt lijkt op het produceren van een rekenkundig bewijs?

We zullen verderop terugkomen op de probllematiek wat mens en machine precies verondersteld worden te doen. Laten we voorlopig simpelweg aannemen dat de machine (definitieve) rekenkundige stellingen opsomt.

Lucas poneert verder dat het formele systeem dat bij een programma hoort heeft een Gödelzin. De Eerste Onvolledigheidsstelling is —tenminste in zijn gebruikelijke formulering— van toepassing op 'rijke' formele systemen, dat wil zeggen formele systemen in de eerste orde predikatenlogica die voldoende rekenkunde omvatten. Het is echter niet zo duidelijk dat het formele systeem van een programma rijk is. Dat systeem is niet een systeem in de eerste orde pre-

[29] Er is een interessant soort dialectische spanning: om te laten zien dat mens en machine verschillend zijn, moeten we er eerst van uit gaan dat ze voldoende hetzelfde zijn. Eén mogelijke route voor Lucas zou ook kunnen ziin om zijn argument de vorm van een reductio te geven: *als* de mechanist gelijk heeft, *dan* is er een *common ground* waarop de prestaties van mens en machine vergeleken kunnen worden. Maar, wat die common ground ook is, de mens is in tenminste één opzicht de machine de baas. Dus heeft de Mechanist ongelijk.

[30] Bovendien is de Gödelzin die bij Lucas zo'n belangrijke rol speelt afhankelijk van het *programma* en niet van *de verzameling stellingen*.

dicatenlogica. Lucas bedoelt ongetwijfeld dat het *door de machine opgesomde systeem* een Gödelzin heeft. Maar het is nog helemaal niet duidelijk dat we de activiteit van de machine willen zien als het opsommen-van-een-systeem. Het opsommen van rekenkundige beweringen is immers niet per se hetzelfde als het opsommen van de bewijzen van een systeem die deze beweringen als conclusies hebben. Gelukkig is er een éénvoudige uitweg. We kunnen gewoon de door de machine opgesomde beweringen zien als axioma's van een systeem in de eerste orde predikaten logica. We kunnen, indien de machine niet de axioma's van de Rekenkunde opsomt, deze zelf toevoegen, zodat het systeem dat we krijgen automatisch rijk wordt. Laten we het zo verkregen systeem voor het gemak *het geassocieerde systeem* noemen. Dit systeem zal eventueel meer stellingen hebben dan de door de machine opgesomde beweringen, maar dat heeft geen effect op de verdere argumentatie. We geven nu onze definitie van het begrip 'consistente machine': *een machine is consistent als het geassocieerde systeem consistent is.* Het geassocieerde systeem heeft een Gödelzin; deze zal —als de machine consistent is— a fortiori niet door de machine worden opgesomd.

4.3 Ad (iii)

Kunnen wij de waarheid van de in stap (ii) te voorschijn getoverde Gödelzin inzien zoals (iii) ons wil laten geloven? *Zo* gesteld is de vraag volstrekt inhoudsloos: er valt niets te zeggen als we niet meer weten over de manier waarop de machine aan ons gepresenteerd wordt. De grote vraag is dus: wat betekent het "gegeven" uit stap (ii)? Er zijn meerdere interpretaties mogelijk. Lucas is het duidelijkst over welke interpretatie hem voor ogen staat in [Luc68]. Ik behandel eerst de meest voor de hand liggende interpretatie van dat "gegeven", die niet Lucas' eigen interpretatie is, maar wel die van sommige commentatoren op [Luc61] —bijvoorbeeld van [Ben67]:

Ik loop over de Oude Gracht. Plotseling komt er een glimmende metalen doos op wieltjes en met flikkerende gekleurde lampjes aanrijden. Nadere beschouwing van de doos leert me dat er aan de voorkant een scherm zit waarop met geregelde tussenpozen rekenkundige stellingen verschijnen: $6+7=13$, voor elk natuurlijk getal n is er een priemgetal p met $p > n$, $1+2+\ldots+n = \frac{1}{2}\cdot n\cdot(n+1)$, ... Kan ik nu de Gödelzin van het geassocieerde systeem bepalen?

Nee, hiervoor heb ik het programma van de machine nodig. Dat programma is immers essentieel voor de beschrijving van het geassocieerde systeem. We moeten dus ons scenario uitbreiden. Een breed grijnzende meneer komt op me toestappen. "Aardig machientje, niet?" Hij laat me het vele vellen beslaande programma zien. Nu kan ik de Gödelzin van het geassocieerde systeem bepalen —in ieder geval in principe: de kans is groot dat ik voor het opschrijven van de Gödelzin meer symbolen nodig heb dan er elementaire deeltjes zijn in het Universum ... Nu is de vraag: kan ik de waarheid van die Gödelzin bepalen? De Gödelzin is waar precies dan als het geassocieerde systeem consistent is. Ik moet dus uitmaken of het systeem consistent is.

Dit is echter in het algemeen niet mogelijk, want: (a) stel je voor dat ik kan inzien dat de machine precies de stellingen opsomt die volgen uit de axioma's van

de Rekenkunde plus het extra axioma: voor alle natuurlijke getallen n zijn er priemgetallen p en q zodat $p-q = 2$ en $q > n$ (het Priemtweelingen Vermoeden). Omdat ik geen idee heb of de Rekenkunde deze laatste bewering weerlegt of niet, heb ik ook geen idee of de geassocieerde theorie van deze machine consistent is of niet. Het is onduidelijk of ik, aannemende dat de geassocieerde theorie de facto consistent is, daar ooit achter kan komen.[31] (b) Onder (a) konden we een overzichtelijke axiomatisering vinden van het geassocieerde systeem.[32] Het is echter goed mogelijk dat dit niet kan. Misschien somt de machine braaf de stellingen van de Rekenkunde op, maar zal zij plotseling —zeg: over een miljoen jaar— '$0 = 1$' als output geven.

Zoals we gezien hebben is (iii) onder het naïeve scenario niet te verdedigen. Laten we nu het scenario uit [Luc68] bezien. Volgens Lucas moeten we het "gegeven" interpreteren in de context van de discussie tussen de Mechanist en de Antimechanist. Wat Lucas wil weerleggen is de bewering dat hij een machine is. Daartoe stelt hij zich voor dat er iemand naar hem toekomt met een geweldig dik pak papier en beweert: "Dit is het programma van de Lucas-machine" Nu kan Lucas concluderen dat de machine die door het programma wordt gespecificeerd consistent is. Het programma is volgens de Mechanist immers het programma van Lucas en Lucas weet dat in ieder geval hijzelf consistent is. Maar weet Lucas eigenlijk wel van zichzelf dat hij consistent is? Lucas zegt (in [Luc61], herdrukt in [And64], zie p53):

> "The fact that we are all sometimes inconsistent cannot be gainsaid, but from this it does not follow that we are tantamount to inconsistent systems. Our inconsistencies are mistakes rather than set policies. They correspond to the occasional malfunctioning of a machine, not to its normal scheme of operations. Witness to this that we eschew inconsistencies when we recognize them for what they are. If we really were inconsistent machines, we should remain content with our inconsistencies, and would happily affirm both halves of a contradiction."

Met andere woorden: mensen zijn in essentie zelfcorrigerende wezens en derhalve per definitie consistent.[33] Lucas is dus consistent en zo moet ook —op grond van de door de Mechanist zelf verstrekte informatie— de door het programma

[31] Een mogelijke tegenwerping tegen het argument zoals hier gepresenteerd is: het gaat er natuurlijk niet om wat ik hier en nu in kan zien, maar om wat ik in principe in kan zien, abstraherend van beperkingen van tijd, geheugenruimte enzovoorts. De principiële situatie voor het geval van de Rekenkunde plus het Priemtweelingen Vermoeden is als volgt: *als* de Rekenkunde plus het Priemtweelingen Vermoeden inconsistent is, dan kunnen wij daar in principe achter komen. Is deze theorie echter consistent, dan is het onduidelijk of wij daar, *zelfs in principe*, achter kunnen komen. De argumentatie onder (a), (b) kan overigens nog wat versterkt worden met metamathematische argumenten, zie bijvoorbeeld: [Bow82].

[32] Deze axiomatisering is volgens onze strikte definitie niet de officiële axiomatisering van het geassocieerde systeem.

[33] Ik denk overigens dat de vergelijking tussen "mistakes" en "the occasional malfunctioning of a machine" niet altijd opgaat: het is op z'n minst vreemd te zeggen dat, bijvoorbeeld, Frege, die een inconsistent systeem bedacht, last had van een storing in zijn essentieel menselijke functioneren.

van de Mechanist gedefinieerde machine consistent zijn. Nu zijn we klaar: Lucas kan de bij het hem gepresenteerde programma horende Gödelzin bepalen. De machine die door het programma gespecificeerd wordt is consistent. Dus Lucas kan deze Gödelzin als waar asserteren, terwijl de machine deze niet zal kunnen opsommen. Ergo, Lucas is niet die machine.

De these dat een zeker programma het programma van Lucas is, is een absurd sterke mechanistische bewering. De argumentatie in deze vorm richt zich tegen een identiteitstheorie die waarschijnlijk op flauwe 'taalkundige' gronden al niet juist kan zijn. Lucas' argumentatie zou echter ook werken —even aannemende dat deze überhaupt werkt— als we ons zouden beperken tot de these dat een zekere machine dezelfde rekenkundige capaciteiten heeft als Lucas. Voor de weerlegging van deze zwakkere these kunnen we het Lucasiaanse scenario ietwat veranderen: nu spreekt de breed grijnzende meneer Lucas aan op de Oude Gracht. Hij stelt zichzelf voor als de Mechanist. Trots toont hij zijn machine en het bijbehorende programma. Hij beweert pertinent dat deze machine in rekenkundig opzicht hetzelfde presteert als Lucas. Lucas poneert op grond van de hierboven beschreven redenering de Gödelzin van de machine en weerlegt daarmee de boude bewering van de nu enigszins sip kijkende meneer.

In mijn bespreking wil ik het gewijzigde Lucasiaanse scenario aanhouden. Als je immers de zwakkere these weerlegt, weerleg je daarmee ook de sterkere. Het is duidelijk dat Lucas tot de waarheid van de Gödelzin van de machine concludeert op grond van de door de Mechanist verstrekte informatie. Het is echter zeer de vraag of Lucas wel het recht heeft de uitingen van de Mechanist als informatie te behandelen. Heeft hij wel reden de Mechanist op zijn woord te geloven? Vreemd genoeg lijkt het argument dat we onder de loupe hebben self-defeating te zijn: Lucas concludeert dat de Gödelzin die hij uit het programma extraheert waar is op grond van de informatie van de Mechanist dat *dat* programma in rekenkundig opzicht hetzelfde presteert als Lucas. Maar uit het verdere argument blijkt dat de Mechanist ongelijk heeft. Lucas had dus van het begin af aan al geen reden om de Mechanist te vertrouwen, en daarmee ook geen reden om te concluderen dat de gevonden Gödelzin waar is. Wat betekent dit nu? Het betekent dat we de onderstelling dat Lucas de Mechanist op zijn woord mag geloven in het argument als hypothese moeten behandelen en dat deze hypothese tot een tegenspraak leidt. De conclusie van het argument is derhalve dat Lucas geen reden *kan* hebben om de Mechanist te vertrouwen. Wil dit zeggen dat de Mechanist dus ook onwaarheid spreekt? Niet per se: het is heel goed denkbaar dat we geen goede reden hebben iemand te vertrouwen, terwijl hij/zij toch de waarheid spreekt.

We komen nu tot wat ik zie als het cruciale bezwaar tegen Lucas' scenario. Lucas verslaat de machine op grond van de door de Mechanist verstrekte informatie. Maar dit is duidelijk geen eerlijke overwinning! De vergelijking is alleen fair als we zowel Lucas als de machine dezelfde informatie verschaffen. Maar als we de machine informatie geven kan het geassocieerde systeem veranderen[34] en

[34]We zouden natuurlijk kunnen proberen te betogen, dat Lucas als hij nieuwe informatie krijgt dezelfde blijft, maar dat een werkende machine bij toevoer van informatie essentieel

hoeft de Gödelzin van het geassocieerde systeem zoals het eerst was niet hetzelfde te zijn als de Gödelzin van het geassocieerde systeem van de geïnformeerde machine. Het is heel goed denkbaar dat de veranderde machine net als Lucas de Gödelzin van het eerdere geassocieerde systeem zal kunnen produceren. Volgens dit bezwaar zijn (ii) en (iii) onder het Lucasiaanse scenario verward en verwarrend omdat daar ten onrechte machines worden behandeld alsof ze niet het vermogen zouden hebben informatie op te nemen, terwijl dit vermogen — terecht— wel aan mensen wordt toegeschreven. In het bijzonder is de Gödelzin waarvan sprake is in (ii) en (iii) de Gödelzin van de ongeïnformeerde machine, maar zijn de 'wij' uit (iii) wel geïnformeerd.

We kunnen de structuur van de argumentatie hier illustreren met een puur machinaal voorbeeld. Stel je twee machines voor: de één, zeg P, somt de Peano Rekenkunde op. De ander, zeg F, implementeert de Feferman Rekenkunde. We zullen in dit voorbeeld niet naar de geassocieerde systemen kijken, maar rechtstreeks naar de geïmplementeerde systemen. Zoals we gezien hebben produceren beide machines precies dezelfde stellingen. Beide machines hebben een invoerpoort via welke we rekenkundige beweringen kunnen inlezen. Deze beweringen worden dan aan de axioma's van de machines toegevoegd.[a] De uitspraak "F somt de zelfde stellingen op als P" kan in de taal van de Rekenkunde vertaald worden. Deze uitspraak is equivalent met de bewering dat de Rekenkunde consistent is! Geen van beide machines zal deze uitspraak kunnen bewijzen. Beide machines zullen precies hetzelfde reageren, wanneer we de uitspraak via de invoerpoort als axioma toevoegen. Natuurlijk zal F —de machine die in ons voorbeeld correspondeert met Lucas— de Gödelzin van de Rekenkunde kunnen bewijzen, als we aan F bovengenoemde bewering toevoegen. Maar het is evengoed zo dat P de Gödelzin van Fefermans systeem kan bewijzen, als we aan P de informatie geven dat F en P in hun onveranderde vorm dezelfde stellingen opsommen.

[a] Er zijn verschillende interpretaties van wat het uitbreiden van F betekent: is het nieuwe axioma wel of niet onderheving aan de backtracking procedure? Gelukkig doet het er voor ons voorbeeld niet toe welke interpretatie we kiezen.

We hebben betoogd dat als Lucas in staat is op grond van de door de Mechanist verstrekte informatie de waarheid van de Gödelzin van de machine te bepalen,

verandert, 'een andere machine wordt'. Dit lijkt mij weinig plausibel. Hoe het ook zij, als Lucas' Argument afhankelijk zou zijn van deze manoeuvre, dan zou het in feite niets anders zijn dan een flauwe taalkundige truc.

dat dan nog niet duidelijk is dat de machine met die informatie niet hetzelfde kan presteren. Dit is voldoende om Lucas' Argument te ontkrachten. Het is echter toch interessant ons af te vragen of Lucas, *als we hem de informatie van de Mechanist gunnen*, inderdaad de waarheid van de Gödelzin van de machine kan bepalen. Dit kan als we de machine opvatten als een apparaat dat stellingen definitief poneert. Maar waarom is dit nodig? Waarom kunnen we een machine niet zien als een zelfcorrigerend apparaat? Lucas probeert redenen te geven waarom dit niet zou kunnen (zie bijvoorbeeld [And64], pagina 53–55): mechanische zelfcorrectie procedures zijn volgens hem wel mogelijk, maar inherent onredelijk: als wij op een tegenspraak stuiten verwerpen we niet zomaar wat axioma's op grond van een vast selectiecriterium; nee, wij analyseren wat er aan de hand is en komen zonodig tot hele nieuwe axioma's, desnoods ook in een nieuwe taal. Ik denk dat Lucas hier gewoon de machinale mogelijkheden onderschat: het is volstrekt onduidelijk wat voor slimme correctieprocedures er niet mogelijk zijn. Je zou je zelfs kunnen voorstellen dat de machine haar zelfcorrectie-protocol bijstelt in het licht van nieuwverkregen informatie![35] Bovendien: als Lucas hier gelijk zou hebben, dan kunnen we Lucas' Argument verder wel missen. Immers, mensen zijn op redelijke manier zelfcorrigerend, machines niet. Dus mensen zijn geen machines. Anders gezegd: als een premisse voor Lucas argument een substantieel verschil is tussen mensen en machines, dan verandert dit argument in een *petitio principii*.

Laten we bezien wat er gebeurt met het Lucasiaanse scenario als we de door de Mechanist gepresenteerde machine opvatten als zelfcorrigerend. Laten we ons voorstellen dat de machine in principe alle gevolgen in de eerste orde predicaten logica van de door haar opgesomde beweringen controleert en dat zij als ze op een tegenspraak stuit bepaalde veronderstellingen —waaronder enige die essentieel zijn bij de afleiding van die tegenspraak— zal intrekken. We nemen nu als axioma's voor het geassocieerde systeem in nieuwe zin de door de machine opgesomde stellingen die nooit worden ingetrokken. Het is éénvoudig in te zien dat het geassocieerde systeem consistent zal zijn; we hebben de mededeling dat Lucas en de machine dezelfde stellingen opsommen helemaal niet nodig om tot die conclusie te komen. De waarheid van de Gödelzin van het geassocieerde systeem volgt echter niet uit de consistentie van de machine: immers dit systeem is geen formeel systeem in de gebruikelijke zin, omdat zijn axioma's de stabiele beweringen zijn van een zelfcorrigererende procedure. Lucas kan dus langs deze weg niet tot de waarheid van deze Gödelzin concluderen!

[35] Een punt dat niet onvermeld mag blijven is dat menselijke zelfcorrectie gericht is op waarheid. Inconsistentie is alleen maar *één mogelijke indicator* van onwaarheid. Een probleem in dit verband is dat de onwaarheid van een systeem —anders dan de inconsistentie— niet in principe herkenbaar hoeft te zijn.

> Bezie de machine F die de Feferman Rekenkunde opsomt. De vertaling in de taal van de Rekenkunde van: "De Feferman Rekenkunde is consistent.", zal door de machine opgesomd worden. De Gödelzin van Feferman Rekenkunde zal niet opgesomd worden. Om tot de waarheid van deze zin te concluderen is het inzicht in de consistentie van Fefermans systeem niet toereikend: we moeten weten dat Peano Rekenkunde consistent is!

De slotsom is: hoe het ook zij, of de door de Mechanist gepresenteerde machine nu als zelfcorrigerend gezien wordt of niet, het is niet duidelijk dat Lucas met de door de Mechanist gepresenteerde informatie meer kan doen dan de machine.

4.4 Ad (iv)

De conclusie van de argumentatie is (iv): machines vormen geen adequaat model van de menselijke geest. Aangezien de essentiële stap (iii) in geen van de beschouwde scenario's is vol te houden, kunnen we niet tot (iv) concluderen.

Nota bene: het enige wat we hebben aangetoond is dat een bepaald argument tegen het Mechanisme faalt. Er volgt natuurlijk niet daaruit dat het Mechanisme juist is.

Wat denk ik nu zelf? Is het Mechanisme in één of andere vorm waar? Ik kan me eerlijk gezegd niet aan het gevoel onttrekken dat we eigenlijk niet goed weten wat de verschillende mechanistische thesen betekenen. Het zou me niets verbazen als de verdere ontwikkeling van de Kunstmatige Intelligentie onze kijk op de inhoud van deze thesen ingrijpend verandert. Eén wel zeer problematisch aspect van het Mechanisme zoals dat in Lucas' Argument naar voren komt is het abstraherend afzien van beperkingen aan tijd en geheugenruimte. Ik meen dat wij —tenminste in de context van de mens-machine problematiek— niet goed begrijpen welke rol deze abstractie speelt.

Tot besluit, wil ik Kreisel het laatste woord laten: "[. . .] wide-eyed enthousiasts of A.I. or, better, of digital intelligence, and indignant critics never give thought to the possibility that we know enough already to refute their rhetoric; but, presumably, not enough to settle any significant issue." (Zie [Kre87].)

5 Nawoord 2004: niets nieuws onder de zon

De bovenstaande tekst is een licht gewijzigde versie van [Vis86]. Zoals te verwachten was, zijn er sindsdien nieuwe artikelen verschenen over het adapteren van Gödelstijl argumentatie ter weerlegging van het Mechanisme. Ten eerste is er een retrospectief artikel van Lucas: [Luc96]. Verder is de discussie nieuw leven ingeblazen door twee flamboyante bijdragen van Sir Roger Penrose: [Pen89],

[Pen94]. Reacties op het werk van Penrose zijn o.a.: [Dav90], [Dav93], [Det95], [Fef95], [Lin01], [Pud99], [Ten01] en [Wri95]. Feferman's kritische bespreking [Fef95] van [Pen94] is ook te vinden op:

⟨`http://psyche.cs.monash.edu.au/v2/psyche-2-07-feferman.html`⟩.

Penrose reageert op sommige kritiek in [Pen95].

Ik denk dat de latere discussie voornamelijk herhaling van zetten is. Weliswaar geeft Penrose ook een argument dat meer verwant is aan de paradox van absolute bewijsbaarheid, die ik hierboven ook besproken heb, maar, als je deze lijn van argumentatie volgt, probeer je conclusies te treken uit een semantische paradox. Uit de bewijsbaarheidsparadox kunnen we evenmin iets substantieels concluderen over mensen en machines als uit de leugenaarsparadox.

Referenties

[And64] A.R. Anderson. *Minds and Machines*. Prentice-Hall, Englewood Cliffs, 1964.

[Ben67] P. Benacerraf. God, the Devil, and Gödel. *The Monist*, 51:9–32, 1967.

[BJ74] G.S. Boolos and R. Jeffrey. *Computability and Logic*. Cambridge University Press, Cambridge, New York, 1974.

[Bow82] G.L. Bowie. Lucas' number is finally up. *Journal of Philosophical Logic*, 11:279–285, 1982.

[C+72] J.N. Crossley et al. *What is mathematical logic?* Oxford University Press, Oxford, 1972.

[Chi72] C. Chihara. On alleged refutations of mechanism using Gödel's incompleteness results. *Journal of Philosophy*, 69:507–526, 1972.

[Dal04] D. van Dalen. *Logic and Structure*. Berlijn, Springer, 2004. Alleen de 4e editie bevat de beoogde behandeling van de onvolledigheidsstellingen.

[Dav90] M. Davis. Is mathematical insight algorithmic? *Behavioral and Brain Sciences*, 13:659–660, 1990.

[Dav93] M. Davis. How subtle is Gödel's theorem? More on Roger Penrose. *Behavioral and Brain Sciences*, 16:611–612, 1993.

[Daw97] John W. Dawson. *Logical Dilemmas, the life and work of Kurt Gödel*. A K Peters, Wellesley, 1997.

[Det95] M. Detlefsen. Wright on the non-mechanizability of intuitionist reasoning. *Philosophia Mathematica*, 3:103–118, 1995.

[FDK+86] Solomon Feferman, John W. Dawson, Stephen C. Kleene, Gregory H. Moore, Robert M. Solovay, and Jean van Heijenoort, editors. *Kurt Gödel, Collected Works, Volume I.* Oxford University Press, 1986.

[Fef60] S. Feferman. Arithmetization of metamathematics in a general setting. *Fundamenta Mathematicae*, 49:35–92, 1960.

[Fef62] S. Feferman. Transfinite recursive progressions of axiomatic theories. *Journal of Symbolic Logic*, 27:383–390, 1962.

[Fef95] S. Feferman. Penrose's Gödelian argument: A review of Shadows of Mind, by Roger Penrose. *Psyche*, 2(7), 1995.

[Fod81] J.A. Fodor. The mind-body problem. *Scientific American*, pages 124–132, January, 1981.

[Hof79] D.R. Hofstadter. *Gödel, Esher, Bach: an eternal golden Braid*. Basic Books, Inc., New York, 1979.

[Kre87] G. Kreisel. Gödel's excursions into Intuitionistic Logic. In P. Weingartner and L. Schmetterer, editors, *Gödel Remembered*, pages 65–179. Bibliopolis, Napels, 1987.

[Lin01] P. Lindström. Penrose's new argument. *Journal of Philosophical Logic*, 30:241–250, 2001.

[Luc61] J.R. Lucas. Minds, Machines and Gödel. *Philosophy*, 36:120–124, 1961.

[Luc68] J.R. Lucas. Satan stultified: a rejoinder to Paul Benacerraf. *The Monist*, 52:145–158, 1968.

[Luc96] J. R. Lucas. Minds, machines, and Gödel: a retrospect. In P.J.R. Millican and A. Clark, editors, *Machines and Thought. The Legacy of Alan Turing*, volume 1, pages 103–124. Oxford University Press, Oxford, 1996.

[NN58] E. Nagel and J.R. Newman. *Gödel's Proof.* New York University Press, New York, 1958. Nederlandse vertaling: De stelling van Gödel. Aula Paperback 136, 1986.

[Pen89] R. Penrose. *The Emperor's New Mind: Concerning Computers, Minds, and the Laws of Physics*. Oxford University Press, New York, 1989.

[Pen94] R. Penrose. *Shadows of the Mind: A Search for the Missing Science of Consciousness*. Oxford University Press, New York, 1994.

[Pen95] R. Penrose. Beyond the doubting of a shadow: A reply to commentaries of shadows of the mind. *Psyche*, 2, 1995.

[Pud99] P. Pudlák. A note on applicability of the incompleteness theorem to human mind. *Annals of Pure and Applied Logic*, 96:335–342, 1999.

[Smu61] R.M. Smullyan. *Theory of formal systems.* Princeton University Press, Princeton, 1961.

[Ten01] N. Tennant. On Turing machines knowing their own Gödel-sentences. *Philosophia Mathematica*, 9:72–79, 2001.

[Vis86] A. Visser. Kunnen wij elke machine verslaan? Beschouwingen rondom Lucas' Argument. In P. Hagoort and R. Maessen, editors, *Geest, computer, kunst*, pages 150–181. Grafiet, Amsterdam, 1986.

[Wan74] Hao Wang. *From Mathematics to Philosophy.* Routledge & Kegan Paul, London, 1974.

[Wan87] Hao Wang. *Reflections on Kurt Gödel.* MIT Press, Cambridge Mass., 1987.

[Web83] J.C. Webb. Gödel's Theorems and Church's Thesis. In R.S. Cohen and M.W. Wartofsky, editors, *Language, Logic, and Method*, pages 309–353. Reidel, Dordrecht, 1983.

[Wri95] C. Wright. Intuitionists are not (Turing) machines. *Philosophia Mathematica*, 3:86–102, 1995.