Seminarium
Hypercomputation, Introductie en Filosofie

Jeroen Broekhuizen, Christian Gilissen, Maurice Samulski

August 23, 2005

Contents

List of Tables

List of Figures

1 Introductie

Dit verslag is geschreven naar aanleiding van de cursus "Seminarium" van de opleiding informatica aan de Radboud Universiteit. Merk op dat dit verslag in velerlei gevallen vertalingen bevat van Engelse termen die de besproken concepten weliswaar benaderen, maar toch niet volledig nauwkeurig vastleggen. Of anders gezegd: er is bij het vertalen van het Engels naar het Nederlands nog wel eens wat van de oorspronkelijke betekenis verloren gegaan.

2 Probleemstelling

Wat is hypercomputation en hoe past hypercomputation binnen het vakgebied informatica.

3 Pre-Hypercomputation

3.1 Geschiedenis

De belangrijkste persoon die geleidt heeft tot hypercomputing is de ons allen welbekende Alan Turing, geboren in 1912 en afgestudeerd aan de King's College universiteit in Cambridge. Hij is het meest bekend geworden om de uitvinding van zijn Turing Machines. Afgezien daarvan heeft Alan nog veel meer uitvindingen gedaan en wiskundige theorien opgesteld. De belangrijkste hiervan, die mede aanzet hebben gegeven aan hypercomputing, zullen in dit stuk aanbod komen.

3.2 Turing Machines

Alan Turing is vooral bekend als uitvinder van zijn turing machines. Deze had hij gemaakt om te proberen algoritmen te formaliseren. In zijn artikel hebben ze het over effectief berekenen, waar ik later op terug zal komen.

Algoritme. *Systematische procedure die - in een eindig aantal stappen - het antwoord geeft op een vraag of de oplossing voor een probleem.*

Deze definitie is lang voordat computers bestonden al opgesteld, en heeft menig wiskundigen bezig gehouden. Het kan terug geleid worden tot in de 9e eeuw waarin de Arabische wiskundige Al-Koarizmi zich hier voor het eerst mee bezig hield. Algoritmische berekeningen hebben te maken met het berekenen van algoritmen.

Algorithmische Berekening. *Een berekening die in een gesloten omgeving uitvoert word waarbij een eindige invoer, bepaalt aan de start van de berekening, omgezet wordt naar een eindige uitvoer die beschikbaar is aan het einde van de berekening, binnen een eindige hoeveelheid tijd.*

De eigenschappen van de Turing machines komen goed overeen met bovenstaande definitie van algoritmische berekeningen; beide worden in een besloten omgeving uitgevoerd, hebben beide eindige hoeveelheid resources (tijd en geheugen) en hebben beide een vast gedrag (alle berekeningen starten in dezelfde configuratie). De Turing machine vormt volgens Eugene Eberbach [1] e.a. de basis van de huidige theoretische computer wetenschappen. Dit komt vooral door de "Strong Turing Thesis" die nog in veel leerboeken aangehaald wordt.

Strong Turing Thesis. *Een Turing machine kan alles wat een echte computer ook kan.[2]*

Veel mensen weten echter niet dat Turing het hier waarschijnlijk niet mee eens was geweest. Hij zag de Turing machine niet als een machine die alles omvatte. Tegenwoordig weet men ook dat dit niet zo is en dat ander modellen nodig zijn om bepaalde berekeningen en problemen op te kunnen lossen.

3.3 Entscheidungsproblem

David Hilbert heeft in 1918 het alom bekende wiskundige Entscheidungsproblem ('beslissings probleem') opgesteld. Hilberts vraag was of het mogelijk is om menselijk redeneren te reduceren is tot algoritmische expressies[3].
Alan was zeer genteresseerd in dit probleem en begon meteen met het opzetten van een bewijs dat dit niet mogelijk zou zijn. In 1931 werd echter al door Gödel bewezen dat het niet mogelijk was. Gödel toonde aan dat er voor elke formele theorie er altijd wel niet beslisbare theorieën zijn.
Het bewijs dat Turing in zijn artikel "On Computable Numbers with an Application to the Entscheidungsproblem" in 1936 publiceerde maakte hij gebruik van een zogenaamde 'automatische machine'. Turing toonde aan dat zelfs deze machine niet alle functies kon berekenen. Hij bewees dat de nu welbekende "halting problem" niet beslisbaar is. Het halting probleem is als volgt te stellen: gegeven een willekeurige Turing machine M met input alfabet Σ en een string w in $\Sigma*$, zal de berekening van M op input w stoppen?[4] Het is te bewijzen door middel van een tegenstelling dat er geen algoritme is die dat kan, maar dat valt buiten het bestek van dit verslag.
De a-machine had de volgende attributen:

- Eendimensionale wisbare tape van oneindige lengte waarop in elke cel symbolen gezet kunnen worden.
- Een schrijf- en leeskop waarmee over de tape genavigeerd kan worden. Deze kop kan een symbool op de huidige plek schrijven en/of lezen.
- Controle mechanisme die bijhoudt in welke state de machine zich bevindt waarvan er eindig zijn.
- Een overgangstabel waaruit de nieuwe state gelezen kan worden voor een gegeven symbool onder de leeskop en de huidige state.

Aan het begin van een berekening staat de machine in de begin toestand (deze is voor iedere berekening in dezelfde machine hetzelfde). Elke stap in de machine leidt tot het lezen van het symbool onder de leeskop. Aan de hand van de overgangstabel bepaald het controle mechanisme wat de nieuwe state is. Het schrijft dan een nieuwe waarde op de tape en gaat naar de nieuwe state. De berekening eindigt wanneer de machine in een zogenaamde "halting" state is aangekomen. Turing geeft aan dat elke machine die zelfstandig compleet kan aan de hand van zijn configuratie kan bepalen naar welke state hij moet vanuit een willekeurige andere state dit een automatische machine is[5]. We kennen deze machines vandaag de dag als Turing Machines. De eigenschappen van de Turing Machine (TM) sluiten zeer nou aan het modeleren van wiskundige berekeningen (gesloten omgeving, eindige hoeveelheid tijd en geheugen en alle TM's starten in dezelfde initile state. De Turing Machines worden nu nog steeds veel gebruikt voor het modeleren van berekeningen.

3.4 Church-Turing Thesis

Zoals hierboven al aangegeven was, zou de Strong Turing Thesis zeer waarschijnlijk door Turing afgekeurd worden. Hij zag de Turing Machine niet als een alles omvattend model. In zijn leven heeft hij dan ook verschillende methoden en machines ontworpen die bepaalde problemen op kunnen lossen (enkele hiervan zullen in een later deel uitvoeriger behandeld worden).
Verschillende mensen waren rond 1930 op zoek naar het bewijzen van Gilberts Entscheidigungsproblem. Dit leverde een aantal nieuwe classes van functies op, waaronder de recursieve functies (Gödel), λ-calculus (Church) en een derde classe TM berekenbare functies (Turing). Zowel Church als Turing waren opzoek naar manieren om berekeningen effectief uit te rekenen. Hierbij doelden ze eigenlijk op mechanische wijze van uitrekenen, maar hebben dat niet compleet uitgewerkt. De Turing Thesis houdt in dat voor elke wiskundige functie, waarvan een effectieve berekeningsmethode bekend is, deze door een Turing Machine berekent kan worden. Door de jaren heen is deze thesis wel bekend geworden en door verschillende wiskundigen onder de loop genomen

maar is (en kan) niet bewezen worden. Om dit te bewijzen zal eerst een formele definitie gegeven moeten worden aan (de niet formele) berekenbaarheid. Dit heeft weer tot gevolg dat deze nieuwe definitie bewezen moet worden, etc. De volgende punten zijn getuigenissen voor de validiteit van de thesis[6]:

- Turing's argument zelf,
- er zijn nog geen tegenvoorbeelden gevonden, ondanks de hoeveelheid werk die er ingestoken is en
- andere onafhankelijke modellen zijn ontworpen voor effectieve berekenen, waarvan later is bewezen dat deze equivalent waren aan de Turing machine.

Niet veel later na zijn thesis bewees Turing dat zijn op Turing machines gebaseerde berekeningen equivalent waren aan Church's λ-definities (hij had al eerder bepaald dat zijn TM's equivalent waren aan de recursieve functies). De oneindige lengte van de tape speelt een zeer belangrijke rol in deze conclusie. In het geval dat de tape eindig is, zijn er dus ook maar eindig veel configuraties mogelijk. De Turing machine zou in dat geval gereduceerd worden tot een Finite State Machine (FSM) die reguliere talen aan kan. Het is bekend dat regulieren talen minder expressief zijn dan talen die TM berekenbaar zijn.
De equivalentie tussen de hierboven genoemde klassen (recursieve functies, lambda definieerbaarheid en TM berekenbaar) werd gebruikt als een bevestiging dat het idee van effectief berekenen eindelijk formeel was. Hierdoor is de Church-Turing Thesis ontstaan. Deze thesis kan gegeven worden in de woorden van Turing[7]:

Church-Turing Thesis. *Every function which would naturally be regarded as computable can be computed by a Turing machine.*

Dit houdt in dat elke functie, die beschouwt wordt als berekenbaar, door een Turing machine berekend kan worden. Voor deze thesis zijn dezelfde argumenten te gebruiken als voor de Turing thesis (zie hierboven). Vele andere machines voor effectieve berekeningen zijn voorgesteld, waaronder de Markov algoritmes, register machines e.a. Voor al deze systemen is aangetoond dat ze essentieel dezelfde functies berekenen als de Turing machine; deze systemen worden Turing compleet genoemd. Hierdoor wordt nu in het algemeen aangenomen dat de Church-Turing thesis correct is. Afgezien van het feit dat de thesis op functies over integers gebaseerd is, heeft hij toch veel invloed gehad (nu nog steeds) op het gebied van computer wetenschappen. Het heeft zelfs een centrale rol gespeeld door zijn robuuste opzet in de oprichting van deze wetenschappen. De thesis is ook relatief eenvoudig uit te breiden tot functies over strings, aangezien strings door rijen van integers gerepresenteerd kunnen worden. Helaas is het niet uit te breiden naar andere typen van berekeningen, zoals functies over real's.

3.5 Turing's Interactie Machines

In Turing's artikel uit 1936 had hij niet alleen de automatische machines gentroduceerd. In hetzelfde artikel had Turing nog een ander model voor berekenen voorgesteld: de keuze machine (c-machine). Turing introduceerde de c-machine als een alternatieve manier van berekenen, en wel een met interactie. De automatische machine opereerde in een 'gesloten doos' omgeving, waarbij geen interactie met de omgeving mogelijk was. De nieuwe c-machine kan daar in tegen met een operator (b.v. een mens) communiceren. Het is slechts gedeeltelijk specificeert in zijn configuratie. In sommige configuraties stopt de machine en wacht op een keuze van de externe operator. Turings doel in zijn artikel was de onbeslisbaarheid van het Entscheidigungsproblem te bewijzen. Hij heeft nooit deze keuze machine formeel gedefinieerd, aangezien de a-machines het al mogelijk maakten zijn doel te bereiken.
Enkele jaren later introduceerde Turing de Oracle machine. Velen geloven dat dit de formele specificatie is van de keuze machine, gezien de gelijkenissen tussen beide machines. In de oude Griekse tijd werden orakels vaak geraadpleegd voor advies. Ze werden gezien als mensen met toegang tot

verborgen kennis die rechtstreeks van de goden afkwam. Net zoals deze Griekse orakels hebben Turing's orakels ook toegang supermenselijke kennis en zijn representeren niet berekenbare informatie verkregen van buiten het systeem (de TM). Turing sloot hierbij de mogelijkheid duidelijk uit dat een orakel een effectieve berekenings entiteit is: *"we shall not go any further into the nature of this oracle apart from saying that it cannot be a machine"*. Formeel kan een orakel beschreven worden als een set die geraadpleegd kan worden over elke waarde. Het orakel retourneert waar als de waarde in de set zit en anders onwaar. Hierboven sprak ik kort over de gelijkenis met de keuze machine. Dit wordt echter tegen gesproken door het feit dat het orakel een set is. Sets geven door hun definitie altijd hetzelfde antwoord welke aan het begin bepaald wordt, wat bij een menselijke operator niet altijd het geval is (kan afhankelijk van zijn stemming, etc).

3.6 Cryptology & complexity theory

Tijdens de tweede wereld oorlog maakte de Duitse marine gebruik van een machine op hun berichten beveiligd over te kunnen sturen tijdens radio communicatie. Deze machine heette de Enigma. Door Turing's ervaring met cryptologie en zijn kennis over combinatorics, code theorie en statistieken droeg hij aanzienlijk bij aan het breken van de enigma codering. Dit gebeurde op Blechley Park. Nog belangrijker was zijn Turing Bombe, een apparaat die op mechanische wijze het ontcijfering proces uit kon voeren. Het apparaat werd later vervangen door de Kolossus, welke gezien kan worden als 's werelds eerste elektrische computer. Later hield Turing zich op Blechley Park bezig met het oplossen van moeilijke problemen in de multi-level encryptie algoritmen. Hij fabriceerde een methode die nu als een interactieve willekeurige manier tot het breken van een geheimschrift beschreven zou worden. Dit was Turing's directe bijdrage aan cryptologie en indirecte bijdrage aan complexiteit theorie. Als "side-note" wil ik nog even vermelden dat door Turings bijdrage aan het breken van de enigma codering, Turing als een bepalende factor voor het winnen van de oorlog wordt gezien. Doordat zijn resultaten top geheim waren, is dat voor vele jaren onbekend gebleven.

3.7 ACE: general universal computer

Na de oorlog kwam Turing in dienst van het "National Physical Laboratory" in 1945. Hier werkte hij aan de Automatic Computing Engine (ACE). Dit was een naoorlogse poging tot het maken van een computer. Turing heeft hier in deze jaren aan gewerkt, maar kon hem niet afronden aangezien zijn supervisors de specificatie die Turing opstelde te futuristisch vonden. Het plan was om een eerst programmeerbare machine te maken met een generieke opzet. De ACE zou het programma en de data in het geheugen bewaren en van daaruit uitvoeren. Turing wilde feitelijk van de ACE een mechanische Turing machine maken en zei dat ook tijdens een lezing tegen de London Mathematical society[8]. De architectuur van de ACE die Turing voorstelde stond in sterk contrast met de andere computer designs van die tijd. Turing wilde er de snelste machine van maken met een kloksnelheid van 1 microseconde en met het meeste geheugen van 60.000 bits. Ook de 32 general-purpose registers die hij voorstelde was nieuw, met een simpel hardware systeem die de basis rekenkundige en booleaanse functies uit kon rekenen. Andere typen etc moesten volgens Turing uitgeprogrammeerd worden. Helaas kreeg dit design pas in de jaren '80 bekendheid als de RISC architectuur. Turing's visie ging echter ver buiten het doel van het project; uitvoeren van lange moeilijke berekeningen. De administratie van het bedrijf, onder leiding van Sir Charles Darwin, vond het idee te revolutionair en besloten het project stop te zetten. Teleurgesteld verliet Turing het bedrijf in 1948. In de jaren '50 is alsnog een prototype van de ACE gebouwd en later onder de naam DEUCE verkocht.

3.8 Artificial intelligence & life

Voordat Turing bij de universiteit van Manchester in dienst trad, schreef hij zijn laatste rapport over de ACE. In dit rapport kwamen ook andere futuristische ideen aan bod, zoals robots die een wandeling maken door de straat. Daarnaast heeft Turing ook nieuwe modellen voor berekenen

voorgesteld, die hij unorganised machines (u-machines) noemde. Helaas werd ook dit rapport niet gepubliceerd, omdat Sir Darwin het als een schoolse essay zag, niet geschikt voor publicatie. Er waren twee typen u-machines. De eerste type was gebaseerd op booleaanse netwerken, welke onderverdeeld was in het A- en B-type. Ze waren opgebouwd uit twee input NAND poorten (neuronen) en gesynchroniseerd door een globale klok. Bij het A type liggen de verbindingen tussen de neuronen vast, terwijl bij het B type gebruik maakte van modificeerbare switch type interconnecties. Hierdoor zou het B type moeten kunnen leren welke connecties aan en welke uit moeten staan. De andere u-machine was het P type welke op finite state machines gebaseerd was. Deze tapeloze turing machines (dus gereduceerd tot FSM, zie hierboven) hadden een incomplete transitie tabel en twee input kanalen voor interactie: het plezier en de pijn signalen. Bij configuraties met missende transities wordt gebruik gemaakt van de twee input kanalen om te bepalen naar welke nieuwe state gegaan moet worden. Turing was er overtuigd van dat zijn B type u-machine zijn universele turing machine kan simuleren, maar gaf nooit een formeel bewijs. De B type heeft per definitie een eindige hoeveelheid neuronen. Het is onbekend of Turing wist dat er oneindig veel neuronen nodig waren voor de simulatie van een oneindige tape. Speculaties gaan de ronde dat Turing dit wel wist, maar geen problemen voorzag bij de uitbreiding van zijn definities naar de oneindige definities. Door de (veel te) late publicatie en Turing's dood, heeft hij ook in dit gebied geen credit gekregen hoewel zijn ideen wel door anderen succesvol werden gebruikt in gebieden als *neurale netwerken* en *reinforcement learning*.
Turing wordt ook gezien als een grondlegger van de artificile intelligentie. Zijn interesse is terug te leiden tot het rapport over de ACE waarin hij spreekt over 'intelligent' gedrag van toekomstige generatie computers. Hij nam schaken als beginpunt voor zijn studie naar intelligente zoek strategien. Zeer optimistisch over de vooruitgangen in de computer wetenschappen, voorspelde hij dat de eerste computer de mens rond 1957 zou verslaan in schaken. In 1997 lukte het de eerste supercomputer, genaamd Deep Blue, om de wereldkampioen schaken Garry Kasparov te verslaan. Deze computer maakte gebruik van een variant van het Minimax algoritme om de volgende zet te bepalen. De computers zouden de mens niet alleen met schaken te verslaan, maar met elk intelligentie probleem. Turing durfde zelfs zover te gaan om in een bekend artikel[9] van hem zijn bekende test voor intelligentie voor te stellen. Deze test is nu bekend als de *Turing test*. Deze wordt later in dit verslag uitvoerig besproken.

3.9 Hilberts Tenth Problem

In dit laatste deel van de geschiedenis gaan we het hebben over Hilberts Tenth Problem. Hilbert heeft in totaal 23 problemen gedeponeerd bij het International Congress of Mathematicians, waarvan de meeste collecties van problemen inhouden. In dit verslag zullen we ons alleen bezig houden met het 10e probleem:

> 10. Determination of the solvability of a Diophantine equation. Given a diophantine equation with any number of unknown quantities and with rational integral numerical coefficients: To devise a process according to which it can be determined by a finite number of operations whether the equation is solvable in rational integers.[10]

Vrij vertaald houdt dit probleem de vraag naar een universele methode om te herkennen of diophantische vergelijking over integers opgelost kunnen worden in. Dit probleem kan gezien worden als een verzameling van drie deelproblemen:

- Als eerste zijn individuele subproblemen, zo gezegd, gelijksoortig. Elke wordt gerepresenteerd door een diophantische vergelijking,
- Ten tweede zijn er een oneindig hoeveelheid van dit soort problemen.
- Ten derde, hebben wiskundigen voor veel van dit soort vergelijkingen al een oplossing gevonden en hebben bewezen dat een andere deel niet is op te lossen. Elke klasse van vergelijking heeft zijn eigen oplossing methode nodig. In zijn 10e probleem, vroeg Hilbert om een universele methode voor het herkennen van de oplosbaarheid van diophantische vergelijkingen.

Een voorbeeld van een systeem van diophantische vergelijkingen is hieronder gegeven:

$$\begin{aligned} 3x2y - 7y2z3 &= 18 \\ -7y2 + 8z2 &= 0 \end{aligned}$$

De vraag is nu of er een x, y en z bestaan in het integer domein waardoor aan beide vergelijkingen voldaan wordt. Het blijkt zo te zijn dat de vraag of een enkele diophantische vergelijking met meerdere variabelen een oplossing heeft in de natuurlijke nummers. Bovenstaand systeem is dan ook alleen op te lossen als de volgende vergelijking oplosbaar is over de natuurlijke nummers:

$$(3(x_1 - x_2)^2(y_1 - y_2) - 7(y_1 - y_2)^2(z_1 - z_2)^3 - 18)^2 + (-7(y_1 - y_2)^2 + 8(z_1 - z_2)^2)^2 = 0$$

In terminologie van tegenwoordig is Hilberts 10th problem een beslissings probleem waarop een JA of een NEE als antwoord moet komen. Van belang hier is om een universele methode te vinden waarmee elk deelprobleem opgelost kan worden. Yuri Matiyasevich bewees in 1970 dat het probleem onbeslisbaar is aan de hand van zijn stelling: "elke recursieve opsombare set is diophantisch". Een set S is recursief opsombaar als en slechts dan als er een berekenbare functie f bestaat met domein $(f) = S$. Dit betekende dat een set S recursief opsombaar is als gegeven een input integer n, als n in de set S zit het algoritme eventueel stopt, anders loopt het eeuwig door. Yuri maakte gebruik van een ingenieuze truc gebaseerd op Fibonacci nummers om aan te tonen dat oplossingen voor diophantische vergelijkingen exponentieel kunnen stijgen. Eerder werk van onder andere Julia Robinson heeft al aangetoond dat dit voldoende bewijs is om aan te geven dat er geen generiek algoritme kan bestaan die de oplosbaarheid van een diophantische vergelijking beslist.

Het 10e probleem. *Er bestaat geen algoritme die voor een willekeurige diophantische vergelijking beslist of deze oplosbaar is.*

Deze stelling is kan echter in een sterkere vorm opgeschreven worden. Het bewijzen dat een dergelijk algoritme niet bestaat kan gedaan worden met behulp van een contradictie. Ga dan uit van het bestaan van een algoritme en deduceer hier dan de contradictie uit. Het enige wat we dan hebben is de deductie. Voor het 10e probleem kunnen we nog wat meer doen. Het niet bestaan van een algoritme voor dit probleem betekend dat elk gegeven algoritme A (die het probleem op zou lossen) mislukt of een foutief antwoord op zal leveren voor een willekeurige vergelijking:

$$D_A(x_1, \ldots, x_m) = 0$$

Dit betekent dat door dit tegenvoorbeeld het algoritme nooit stopt of zijn uitvoer, als die er al is, foutief is. De sterkere stelling wordt dan ook:

De sterkere stelling van het 10e probleem. *Er is een algoritme die voor een gegeven algoritme A een tegenvoorbeeld produceert als we aannemen dat A Hilberts 10e probleem oplost.*

Algoritme A kan hierbij gerepresenteerd worden in elke standaard vorm; bijvoorbeeld door een Turing machine of een pascal programma.

4 Hypercomputation

4.1 HyperComputation begrippen

Op dit moment is hypercomputation nog erg theoretisch en wordt er veel over gediscussieerd. Een interessant voorbeeld van zo'n discussie zijn de papers van Bringsjord en Zenzen[23]. Bovendien is Hypercomputation zeer breed, het grenst aan de natuurkunde, filosofie en informatica. In de literatuur worden verschillende termen door elkaar heen gebruikt die erg verwarrend kunnen zijn. We geven hier de definities van die termen samen met een korte uitleg[13]:

Super-Turing: Alle vormen van informatie verwerking die een Turing Machine niet kan. [1]

Super-Turing Computation: Werd in de literatuur over neurale netwerken gebruikt om machines te beschrijven met verschillende extra mogelijkheden. (Die meer konden dan een Turing Machine)

HyperComputation: De theorie van berekeningsmethoden van niet-recursieve functies.

Natural Computation: Berekeningen die voorkomen in- of genspireerd zijn door de natuur.[11, 12]

In sommige literatuur wordt wel eens het volgende plaatje gebruikt:

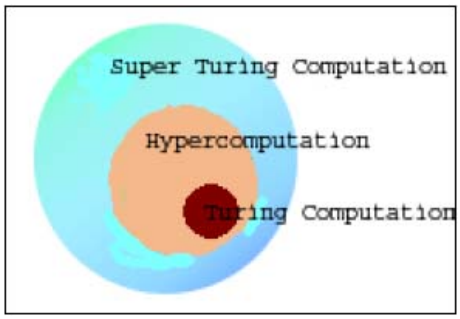

Figure 1: Hypercomputation en Super-Turing [13]

Bij dit plaatje zijn op zijn minst 2 kanttekeningen te plaatsen:

1. Recursieve berekeningen worden hier gezien als een subset van niet-recursieve berekeningen. Naar onze mening is dit onjuist en zijn de beide verzamelingen onderling disjunct. Immers een berekening kan ofwel wel door een Turing Machine uitgevoerd worden, ofwel niet. Niet beide, nog geen.

2. Uit het plaatje blijkt dat Turing berekeningen ook Super-Turing berekeningen zijn. Ook dit is naar onze mening enigszins discutabel. Het bovendien nog maar de vraag of een machine die Super Turing berekeningen kan doen, ook wel Turing berekeningen kan doen.

4.2 Mogelijkheden voor HyperComputation

Uit het werk van Toby Ord[15] komen 5 stellingen die afhankelijk van het feit of ze wel of niet juist zijn verschillende universes scheppen.

A. Alle processen die uitgevoerd kunnen worden door gedealiseerde mathematici zijn te simuleren door Turing Machines.

B. Alle wiskundige harnesable[2] processen in het universum kunnen worden gesimuleerd door Turing Machines.

C. Alle fysieke harnesable processen in het universum kunnen worden gesimuleerd door Turing Machines.

D. Alle processen in het universum kunnen worden gesimuleerd door Turing Machines.

E. Alle formaliseerbare processen zijn simuleerbaar door Turing Machines.

[1] In sommige literatuur ook wel: alle vormen van informatie verwerking die een Turing Machine niet kan en alle vormen van informatie verwerking die een Turing Machine wel kan: dus alle vormen van informatie verwerking!

[2] harnesable in de betekenis van: heeft een eindige mathematische definitie die door middel van de wetenschap kan worden gevonden

Hierin is A. de Church-Turing Thesis, en is E. uiteraard onjuist, wat was bewezen door het Halting probleem. Afhankelijk van het feit of de overige stellingen nog juist zijn, leven we in verschillende universes:

B, C, D Er bestaat geen Hypercomputation in het universum, Turing Machines zijn voldoende om alle processen te simuleren.

B, C Het universum is HyperComputing, maar we kunnen er niet aankomen; We kunnen alleen gebruik maken van de kracht van Turing Machines.

B Het universum is HyperComputing en het is op z'n minst theoretisch mogelijk om hiervan gebruik te maken.

GEEN HyperComputation bestaat, en we kunnen het ook gebruiken.

4.3 Methoden voor Super-Turing berekeningen

Enkele specifieke methoden van Super-Turing berekeningen zijn al in vorige seminaria aan bod gekomen:

- Analoge Computers (Lezing Super-Turing berekeningen: Analoge computers e.d. op 12 november 2004)
- Analoge Neurale netwerken (Lezing Super-Turing berekeningen: Analoge computers e.d. op 12 november 2004)[12, 26]
- Quantum Mechanische systemen (Lezing Super-Turing berekeningen: Quantumberekeningen op 19 november 2004)[24]
- Pulse Computers (Lezing Super-Turing berekeningen: Pulse-berekeningen op 3 december 2004)

Tien D. Kieu[17, 18, 19] schreef al een paper waarin hij bewees hoe een quantum algoritme "Hilbert's tenth problem" kon oplossen. Dit heeft hij later echter moeten aanpassen en vooralsnog gaat het om een algoritme dat vermoedelijk "Hilbert's tenth problem oplost".

4.4 Uitbreidingen van Turing Machines

Enkele uitbreidingen van Turing machines zijn op 17 december 2004 aan bod gekomen in de lezing: Hypercomputation: rare berekeningsmodellen. Enkele van de meest bekende uitbreidingen zijn[1]:

- O-machines: Turing Machines met een Orakel.
- Turing Machines met initiele inscriptie: Turing Machines waarvan de tape bij aanvang is beschreven met oneindig veel tekens.
- Gekoppelde Turing Machines: Turing Machines die hun invoer gaande weg gevoerd krijgen vanuit een externe (niet-recursieve) bron.
- Asynchrone netwerken van Turing Machines: Turing Machines in een netwerk die gebruik maken van een (niet-recursieve) timer functie.
- Error prone Turing Machines: Turing Machines die fouten maken volgens een (niet-recursieve) fout functie.
- Probabilistic Turing Machines: Een Turing machine die vanuit iedere state twee mogelijkheden heeft met gelijke kansen voor elke mogelijkheid.
- Oneindige toestanden Turing Machines: Een Turing Machine met oneindig veel toestanden.

- Versnellende Turing Machines: Turing Machines die iedere stap sneller uitvoeren.
- Oneindige tijd Turing Machines: Uitbreiding van Versnellende Turing Machines.
- Eerlijke niet-deterministische Turing Machines: Niet deterministische Turing Machines waarin een lus niet oneindig vaak herhaald wordt.

De details van iedere machine zijn in de lezing van 17 december behandeld.
Het is duidelijk dat al deze uitbreidingen gebaseerd zijn op slechts 3 concepten:

- Oneindig veel tijd.
- Oneindig veel geheugen
- Een niet-recursieve informatie bron.

5 Filosofie van Hypercomputation

We kijken nu even naar wat filosofie over Hypercomptation.[14]

5.1 Definities

Een supertaak is een gedefinieerd als een oneindige opvolging van acties of operaties die wordt uitgevoerd in een eindig tijdsinterval. De begrippen actie en operatie hebben hier niet perse betrekking op mensen. We karakteriseren het begrip actie namelijk als volgt: We nemen aan dat op ieder willekeurige tijdstip de wereld kan worden beschreven door een reeks S van zinnen. Een actie of operatie die wordt toegepast op de toestand van de wereld resulteert in een verandering van deze toestand. Een willekeurige actie definiëren we aldus als een verandering in de toestand van de wereld. De toestand S verandert dan in een toestand $a(S)$ na de actie a. Hierdoor heeft een actie een begin en een einde, maar dit betekent niet dat hiertussen een eindige tijd verstrijkt.

Sommige schrijvers zijn het niet eens met de gevolgen van deze definitie van actie. Wanneer we inderdaad werkten met het algemene filosofische probleem van verandering, hebben ze misschien gelijk. Maar in dit geval hoeven we ons daar geen zorgen om te maken aangezien in de meeste gevallen van suptertaken, onmiddellijke taken kunnen worden vervangen door taken die een eindige tijd duren, zonder dat daardoor de analyse fundamenteel wordt aangetast.

Er is bovendien een specifiek type supertaak genaamd hypertaken. Een hypertaak is een niet enumereerbare oneindige opvolging van acties of operaties die uitgevoerd wordt in een eindig tijdsinterval. Een supertaak die geen hypertaak is, is een numereerbare oneindige opvolging van acties of operaties die uitgevoerd wordt in een eindig tijdsinterval. Een taak kan gezien worden als een eindige opvolging van acties of operaties die worden uitgevoerd in een eindig tijdsinterval.

5.2 Het filosofische probleem van supertaken

Om een beter inzicht te krijgen in de filosofische problemen van supertaken, bekijken we het verschil tussen taken in het algemeen (een eindige opvolging van acties van het type $(a_1, a_2, a_3, \ldots, a_n, \ldots)$ en een specifiek type supertaken, namelijk supertaken die bestaan uit een oneindige opvolging van acties van het type $(a_1, a_2, a_3, \ldots, a_n)$, die dus van hetzelfde orde type zijn zijn als de orde van de natuurlijke positieve getallen: $1, 2, 4, \ldots, n, \ldots$. We noemen deze supertaken: supertaken van het type w.

In het geval van een taak $T = (a_1, a_2, a_3, \ldots, a_n)$ is het natuurlijk om te zeggen dat T toepasbaar is op een toestand S als

a_1 is toepasbaar is op S,
a_2 toepasbaar is op $a_1(S)$,
a_3 toepasbaar is op $a_2(a_1(S))$,
$\ldots$, en
a_n toepasbaar is op $a_{n-1}(a_{n-2}(\ldots(a_2(a_1(S)))\ldots))$.

De op elkaar volgende toestanden van de wereld relevant voor taak T kan worden gedefinieerd door middel van een eindige opvolging van reeksen van zinnen: $S, a_1(S), a_2(a_1(S)), a_3(a_2(a_1(S))), \ldots, a_n(a_{n-1}(a_{n-2}(\ldots(a_2(a_1(S)))\ldots)))$, wiens laatste term de relevante toestand van de wereld zal beschrijven *na* de uitvoering van taak T. We bekijken nu het geval van een supertaak $T = (a_1, a_2, a_3, \ldots, a_n, \ldots)$. We noemen de taak die de eerste n acties van T uitvoert, T_n. Dus $T_n = (a_1, a_2, a_3, \ldots, a_n)$. Het is nu natuurlijk om te zeggen dat T toepasbaar is op een toestand S als T_n toepasbaar is op S voor ieder natuurlijk getal n. De op elkaar volgende toestanden van de wereld relevant voor supertaak T kunnen worden beschreven door middel van de oneindige opvolging van reeksen van zinnen: $S, T_1(S), T_2(S), \ldots, T_n(S), \ldots.$ Het is nu echter moeilijk om de reeks van zinnen te vinden die de relevante toestand van de wereld beschrijft *na* uitvoering van de supertaak T omdat de oneindige opvolging geen eind term heeft. Er lijkt geen eindige toestand te zijn als resultaat van de toepassing van T op S. Het probleem is nog moeilijker doordat de definitie van supertaak T stelt dat de supertaak wordt uitgevoerd binnen een eindig tijdsinterval. Dus moet er een instantie van tijd $t*$ zijn waar er gesteld kan worden dat de supertaak is afgerond. Het is duidelijk dat de wereld op tijdstip $t*$ in een bepaalde toestand moet zijn, de toestand die het resultaat is van de toepassing van T, maar dat we serieuze moeite hebben om deze toestand te specificeren.

5.3 Supertaak: een vaag begrip

Met onze huidige definities van acties, toestanden als reeksen van zinnen rijst de vraag welke beschrijvingen wel toegelaten mogen worden en welke niet. Hierdoor zijn er supertaken waarvan bijna alle filosofen het er over eens zijn, dat het inderdaad supertaken zijn. Er zijn echter ook supertaken waar de meningen over verschillen.

Een voorbeeld van het eerste geval is Thomson's lamp. Thomson's lamp is een apparaat dat bestaat uit een lamp en een knop met een elektrisch circuit. The knop heeft 2 standen en de lamp is of wel *aan* (de knop staat in de stand 'aan') ofwel *uit* (de knop staat in stand 'uit'). Neem aan dat in het begin (op t = 0:00) de lamp *uit* staat en dat de lamp vanaf dat moment is onderworpen aan de volgende oneindige opvolging van acties: Wanneer de helft van de tijd tot $t*$ = 1:00 voorbij is, voeren we actie a_1 uit, het omzetten van de knop naar stand 'aan', met als resultaat dat de lamp *aan* zal zijn. (a_1 wordt dus uitgevoerd op $t = \frac{1}{2}$; Wanneer de helft van de tijd tussen het uitvoeren van a_1 en $t* = 1$ is verstreken, voeren we actie a_2 uit en zetten de we knop in stand 'uit', waarmee we de lamp uitzetten(a_2 wordt dus uitgevoerd op $t = \frac{1}{2} + \frac{1}{4}$) en zo verder. Op tijdstip $t* = 1$ hebben we een oneindige opvolging van acties uitgevoerd, een supertaak $T = (a_1, a_2, a_3, \ldots, a_n, \ldots)$. Als we alleen kijken naar de veranderingen van de lamp (en niet de knop) en de relevante toestand van de wereld, dan zien we dat onze supertaak slechts twee beschrijvingen van de toestand heeft: {lamp aan} en {lamp uit}.

Een voorbeeld van het tweede geval (supertaken waar men het dus niet over eens is, of het inderdaad wel supertaken zijn) is een proces in één van de vormen van Zeno's dichotomie paradox. Stel dat in het begin (t = 0:00) Achilles bij punt A ($x = 0$) is en dat hij zich verplaatst in een rechte lijn met een constante snelheid $v = 1\,km/h$ naar punt B ($x = 1$), dat op 1 km afstand ligt van punt A. Bovendien veronderstellen we dat Achilles zijn snelheid nooit aanpast. In dit geval kunnen we Achilles zijn rennen zien als een supertaak: Wanneer de helft van de tijd tot $t*$ =1:00 voorbij is, zal Achilles taak a_1, het rennen naar van punt $x = 0$ naar $x = \frac{1}{2}$ hebben uitgevoerd (a_1 is dus uitgevoerd in het tijdsinterval tussen $t = 0$ en $t = \frac{1}{2}$). Wanneer de helft van de tijd vanaf

het einde van taak a_1 tot $t* = 1$ voorbij is, zal Achilles taak a_2 hebben uitgevoerd, het rennen van punt $x = \frac{1}{2}$ tot punt $x = \frac{1}{2} + \frac{1}{4}$, en zo verder. Op tijdstip $t* = 1$ zal Achilles een oneindige opvolging van acties, supertaak $T = (a_1, a_2, a_3, \dots, a_n, \dots)$ hebben uitgevoerd, aangenomen dat we de toestand van de wereld relevant voor de beschrijving van T mogen specificeren op een willekeurig moment door middel van een enkele zin: de zin die Achilles positie op dat moment specificeert.

Verschillende filosofen hebben bezwaar gemaakt tegen deze conclusie met het argument dat in tegenstelling tot Thomson's lamp, het rennen van Achilles niet oneindig veel acties bevat, maar pseudo-acties. Zij zien het rennen van Achilles als niets anders dan het opbreken van één proces in oneindig veel pseudo processen. Sommige filosofen vinden dat de fout zit in het toestaan van alle soorten zinnen voor de beschrijving van de wereld. Zij zijn van mening dat er alleen verschillende type zinnen mogen worden gebruikt voor de beschrijving van de toestand van een wereld die relevant is voor een bepaalde actie (kortom de opvolging van de zinnen moet zo zijn dat er een duidelijk verschil is tussen ieder paar zinnen dat elkaar opvolgt). Zij zijn in ieder geval van mening dat Achilles rennen geen supertaak is.

5.4 De mogelijkheid van supertaken

Bovenstaande problemen zijn uiteraard alleen maar problemen als men het eens is met de gebruikte concepten en formuleringen. Sommige filosofen verwerpen deze problemen omdat ze het gehele concept van een supertaak als problematisch ondervinden. De eerste bekende filosoof die er zo over dacht was Zeno van Elea.

5.4.1 Zeno's Dichotomie Paradox

Wanneer we het probleem van Achilles bekijken beweert Zeno dat Achilles nooit het punt B zal bereiken; immers Achilles moet eerst de halve afstand afleggen, daarna moet hij de helft van de volgende helft afleggen, en zo verder. Er zal voor Achilles altijd een nieuw midden zijn dat hij moet bereiken, waardoor hij dus nooit bij punt B zal aankomen. Volgens Owen (Owen [1957-58]) wilde Zeno hiermee aantonen dat het Universum een eenvoudige globale eenheid is die niet uit verschillende delen bestaat. Hij wil laten zien dat als we dingen gaan opdelen we absurde resultaten krijgen en dat we daarom niet moeten proberen de wereld op te delen. Supertaken zijn dus onmogelijk in een Zenoniaanse wereld.

De dichotomie paradox wordt standaard opgelost door te zeggen dat de opeenvolgende afstanden die Achilles rent - $\frac{1}{2}, \frac{1}{4}, \frac{1}{8}, \dots$ - een oneindige reeks vormen wiens som gelijk is aan 1. Hierdoor zal Achilles dus wel punt B (x = 1) bereiken op t* = 1. Dus is er geen probleem in het opsplitsen in kleinere stukken en is er dus ook geen probleem met het idee van een supertaak. Er kan echter wel een bezwaar worden gemaakt tegen deze standaard oplossing. Het vertelt ons wel waar Achilles zal zijn op ieder tijdstip, maar het vertelt ons niet waar de fout zit in Zeno's beredenering. Bovendien is er nog een bezwaar. Het optellen van de oneindige hoeveelheid sub-afstanden is eigenlijk ook een supertaak. Hierdoor wordt kan men stellen dat deze standaard oplossing aantoont dat supertaken mogelijk zijn, door middel van de aanname van een supertaak. Hier kan een voorstander van de standaard oplossing echter weer tegenin brengen dat er niet oneindig veel getallen worden opgeteld, maar dat de som van de serie slechts de limiet is waartoe de gedeeltelijke (en dus eindige) som nadert. Uiteraard kan men nu echter weer aanvoeren dat dit geen geldig argument is aangezien er gebruik wordt gemaakt van een definitie uit de wiskunde voor oneindige reeksen en dat het niet acceptabel is om empirische stellingen af te leiden uit definities.

5.4.2 De inverse vorm van Zeno's Dichotomie Paradox

Een nog meer omstreden supertaak is de zogenaamde inverse van Zeno's Dichotomie Paradox. Stel dat Achilles zich op tijdstip t = 0:00 bij punt A bevindt en dat hij naar B ($x = 1$) wil. Om

bij B te komen zal hij echter eerste de halve afstand tot B moeten overbruggen. Om echter de halve afstand tot B te overbruggen moet hij eerst de helft van deze halve afstand overbruggen, en zo verder. Om B te bereiken zal Achilles dus een oneindige opvolging van acties, een supertaak $T* = (\ldots, a_n, \ldots a_3, a_2, a_1)$ moeten uitvoeren. Merk op dat T* dezelfde type natuurlijke orden heeft als de negatieve gehele getallen: $\ldots, -n, \ldots, -3, -2, -1$. (Zo'n typen noteren we meestal met de expressie 'w*', en de gerelateerde supertaken noemen we dan ook wel supertaken van het type w*). Als we nu kijken naar de reeksen van zinnen die de wereld beschrijven dan zien die er als volgt uit: $\ldots, a_n(S), \ldots, a_3(S), a_2(S), a_1(S)$. Veel filosofen vinden het onacceptabel dat de begin reeksen niet kunnen worden gespecificeerd en dat we dus de taak die als eerste moet worden uitgevoerd niet kunnen specificeren. Volgens sommigen is dit bewijs genoeg van het feit dat supertaken van het type w* niet mogelijk zijn.

5.4.3 Over Thomson's argumenten tegen supertaken

Thomson was er van overtuigd dat supertaken onmogelijk zijn. Om deze reden bedacht hij Thomson's lamp. Hiermee wilde hij aantonen dat alle supertaken onmogelijk zijn. Beschouwen we namelijk de lamp op tijdstip t = 1:00 en vragen we ons af, is de lamp *aan* of *uit*? Volgens Thomson kan de lamp niet *aan* zijn, want iedere keer nadat de lamp wordt aangezet, wordt de lamp ook weer uitgezet. De lamp kan ook niet *uit* zijn, want ook al is de lamp initieel *uit*, we zetten de lamp daarna aan en iedere volgende keer dat we we lamp uitzetten, zetten we de lamp meteen erna weer *aan*. Dus de lamp is niet *aan*, en ook niet *uit*; maar onze aanname was dat de lamp ofwel *aan*, ofwel *uit* is: een tegenspraak dus. Conclusie de taak behorende bij Thomson's lamp is onmogelijk.

Benacerraf (1962) ontdekte een serieuze tekortkoming in Thomson's argument. Als we kijken naar waarom de lamp op $t* = 1$ *uit* is, dan gebruikt Thomson het argument dat $t* = 1$ afhangt van de voorgaande actie uit de reeks. Echter, $t* = 1$ maakt helemaal geen deel uit van de reeks. Dit is overigens precies dezelfde reden waarom ook de standaard oplossing van Achilles niet werkt. Ook hier wordt er iets geconcludeerd voor een tijdstip 1 met behulp van een reeks, terwijl $t* = 1$ geen deel uit maakt van deze reeks.
Thomson had nog een argument tegen het bestaan van zijn lamp: we kennen de waarde 0 toe aan een lamp die *uit* is, en de waarde 1 als de lamp *aan* is. Aanzetten zien we als er 1 bij optellen en uitzetten als er 1 van afhalen. We kunnen eindtoestand van de lamp dan beschrijven door middel van de oneindige reeks: $1, 1-1, 1-1+1, 1-1+1-1\ldots$. Er kan voor deze serie nooit met behulp van een limiet de som bepaald worden, en om die reden kan ook de lamp op tijdstip $t = 1$ nog de waarde 1 noch de waarde 0 hebben (wat weer in tegenspraak is met de aanname). Benacerraf's antwoord hierop was dat de som van de serie niet perse het antwoord was op de eindtoestand van de lamp. Een eigenschap van de elementen van een reeks hoeft niet perse een eigenschap te zijn van de limiet van die reeks. De partiële sommen van de reeks $0.3 + 0.03, 0.003\ldots$ zijn bijvoorbeeld allemaal kleiner dan $\frac{1}{3}$, maar de limiet waar deze partiële sommen naar neigen is $0.3333\ldots$ en dat is precies $\frac{1}{3}$.
Black (1950-51) gebruikte een soortgelijk argument als Thomson met behulp van een zelf bedachte oneindigheids machine die hij een bal laat verplaatsen. Ook zijn argumenten zijn echter op dezelfde wijze te weerleggen als die van Thomson.

5.4.4 Benacerraf's kritiek op het Dichotomie argument

In bovenstaande gevallen wordt er door Benacerraf gebruik gemaakt van 2 ideeën om de argumenten tegen supertaken onderuit te halen:

1. De toestand van een systeem op een tijdstip $t*$ is geen logisch gevolg van de eerdere toestanden.

2. De eigenschappen die gedeeld worden door de partiële sommen van een serie hoeven niet ook eigenschappen te zijn van de limiet waar deze sommen toe neigen.

Idee 2 kan nog worden omgeschreven naar andere vorm: de eigenschappen die gedeeld worden door de termen van een reeks, hoeven niet perse eigenschappen te zijn van de limiet waar de reeks naar neigt.
We kunnen dit laatste idee gebruiken om Zeno's Dichotomie argument mee te bestrijden. Ook al is het zo dat Achilles op geen enkel tijdstip $t_1, t_2, \ldots t_n$ zijn taak volbracht heeft, dan kan hij toch klaar zijn op tijdstip $t = 1$. Eenzelfde beredenering werkt bij de inverse Dichotomie stelling.
Samengevat zijn er dus geen logische argumenten tegen het uitvoeren van supertaken.

6 De HyperComputation discussie

Er zijn nogal wat standpunten in het HyperComputation debat. We lichten 3 van deze standpunten nader toe.

1. HyperComputation is niet mogelijk
2. HyperComputation is wel mogelijk, maar niet binnen onze huidige Ruimte-Tijd.
3. HyperComputation is wel mogelijk en bestaat zelfs al.

6.1 HyperComputation is niet mogelijk

Er zijn nogal wat critici op het gebied van Hypercomputation. Enkele van de bekendste kritiekpunten op Hypercomputation[21] worden hieronder kort toegelicht:

6.1.1 Fysiek onmogelijk

De meeste HyperComputer machines zijn fysiek onmogelijk:

- Versnellende Turing Machines
- Analoge computers
- Analoge Neurale netwerken

Versnellende Turing Machines
Volgens de huidige natuurkundige wetten (van bijvoorbeeld Newton en Einstein) zijn bovenstaande machines fysiek niet mogelijk. De kleinst meetbare waarde met enige fysieke betekenis is de constante van Planck: ca. $4 \cdot 10^{-35}$ sec. (10^{-20} maal de grootte van een proton) De kleinste meetbare tijd die enige fysieke betekenis heeft is de Planck tijd: de tijd die je nodig hebt om de Planck lengte te overbruggen met de lichtsnelheid: ca. $1.4 \cdot 10^{-43}$ De 143ste stap van een versnellende Turing machine duurt 2^{-143} seconden, wat minder is dan $1.4 \cdot 10^{-43}$. Operaties die zo snel gaan hebben vanuit een fysiek oogpunt geen zin. Zelfs als we de machine $4.5 \cdot 10^9$ jaar de tijd geven, zal de Planck tijd al bij de 200ste stap bereikt zijn.

Analoge computers
Voor analoge computers geldt dat ze niet realistisch zijn in een wereld met ruis, eindige nauwkeurigheid, en quantum effecten.

6.1.2 Empirische betekenisloos

Een ander veelgebruikt argument tegen Hypercomputers is "empirical meaninglessness". Gegeven een hypercomputer, zouden we nooit kunnen bepalen of het inderdaad een hypercomputer is, noch wat voor een hypercomputer het is (wat hij doet). Dit komt omdat we slechts een eindig aantal observaties kunnen doen, met beperkte nauwkeurigheid. Dus de bewering dat een bepaalde machine een hypercomputer is, in plaats van een Turing Machine in empirisch betekenisloos.
Hieronder een voorbeeld van een eenvoudige analoge computer met de argumenten geconcretiseerd.

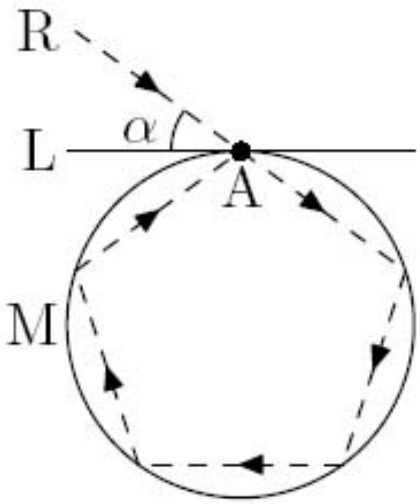

Figure 2: Een analoge computer[21]

Hierbij is M een ronde spiegel met de reflecterende zijde aan de binnenkant. A is een klein gaatje in M met een semi-transparante detector (er mag wel licht in, maar er kan geen licht uit). R is een binnenkomende lichtstraal. α is de hoek tussen R en het horizontale vlak L. Volgens de wetten van de geometrie, wordt de detector bij A alleen geraakt door de lichtstraal als er rationeel getal q bestaat zodanig dat $\alpha = q \cdot \pi$. (In deze figuur, $\alpha = \frac{1}{5} \cdot \pi$) Dus het apparaat bepaald of α een rationeel meervoud is van π. Geen enkele Turing Machine kan zo'n berekening uitvoeren.
Het is duidelijk dat deze computer fysiek niet mogelijk is. Immers A moet oneindig klein zijn en M moet perfect rond zijn. Beide zijn fysiek onmogelijk in deze wereld. Bovendien zouden we nooit zeker kunnen weten dat α inderdaad is wat we willen.

6.1.3 De Beckenstein grens

Stelling: een ruimte met radius r en energie E can slechts een beperkte hoeveelheid informatie I bevatten. Met daarbij $I \leq 2\pi ER/\hbar c \ln 2$ en $\hbar$ de constante van Planck en c de lichtsnelheid
The Beckenstein Grens impliceert dat een Turing Machine met oneindig veel configuraties fysiek niet mogelijk is (in eindige ruimte en begrensde energie).

6.1.4 Conclusies

Ondanks al deze argumenten tegen Hypercomputers zijn de critici het er echter toch over eens dat:

> In short it would (or should) be one of the greatest astonishments of science if the activity of Mother Nature were never to stray beyond the bounds of Turing-machine-computability.

Copeland[16] weerlegt al deze argumenten tegen Hypercomputation, maar doet dat naar onze mening niet erg overtuigend.

6.2 HyperComputation is wel mogelijk, maar niet binnen onze huidige Ruimte-Tijd

Omdat het met de hier geldende wetten fysiek onmogelijk lijkt om bepaalde Hypercomputers te bouwen, zijn wetenschappers hun heil elders gaan zoeken. Namelijk in space time structuren die

zouden kunnen bestaan volgens de algemene relativieit. Malament en Hogarth toonden aan dat in specifieke space-times (namelijk Malament-Hogarth space times) hypercomputation mogelijk is.[20, 22] Hogarth bewees dat het mogelijk is om in zo'n Malament-Hogarth space time ofwel het entscheidungsproblem, of wel het halting probleem op te lossen. Simplistisch is de methode in onderstaande figuur weergegeven.

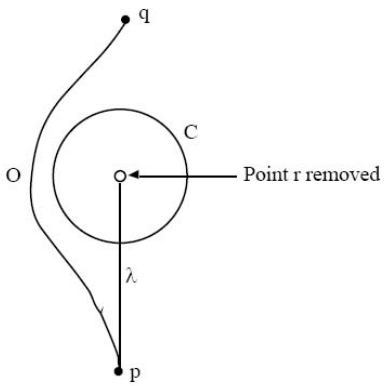

Figure 3: MH Space-Time[22]

Hierbij wordt er wel even van het gemak vanuit gegaan dat een Turing Machine in een dergelijke space time zou kunnen bestaan. Het is echter zeer wel mogelijk dat in dergelijke ruimten materie niet zo kan bestaan als bij ons.

6.3 HyperComputation is wel mogelijk en bestaat zelfs al

Zoals gezegd wordt er in de literatuur nog wel eens slordig omgesprongen met de termen Hypercomputation en Super-Turing.[25] Waar artikelen het veelal over hebben wanneer ze beweren dat Hypercomputation al bestaat zijn Super-Turing berekeningen.
Een voorbeeld hiervan is "Driving-Home-From-Work".[1] Stel we willen een auto programmeren om ons naar huis te rijden. We geven de auto een kaart van de wereld waar alles op staat en maken een algortime dat een route naar huis berekent en uitvoert. Ondanks al deze informatie zal het de auto niet in staat zijn om zelfstandig naar huis te rijden. Dit om de simpele reden dat hij het gedrag van mensen niet kan voorspellen.[3] We kunnen dit probleem dus niet algoritmisch oplossen.

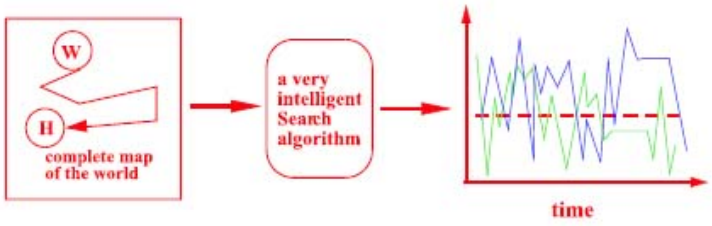

Figure 4: overgenomen van [1]

We breiden nu de auto uit met sensoren en camera's en laten de auto terwijl hij zijn plan uitvoert zijn omgeving registeren en indien gewenst zijn plan aanpassen. Op deze manier kan hij wel zelfstandig naar huis rijden. Dit is een typisch Artificial Intelligence scenario, dat niet door een

[3]Kon hij dat wel, dan betekende dat het gedrag (en dus het lot) van ieder mens ergens vast ligt.

Turing Machine kan worden opgelost. Dit uiteraard om de simpele reden dat een Turing Machine niet gaandeweg nieuwe invoer kan verwerken. Alle invoer ligt aan het begin vast.

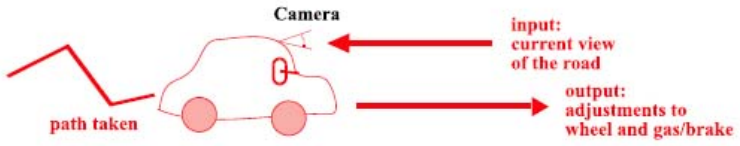

Figure 5: overgenomen van [1]

Enkele andere voorbeelden van Super-Turing berekeningen in het echte leven zijn:

- Distributed Client/Server computation
- Mobile robotics
- Evolutionary computation

7 Hyper-berekenbaarheid is onweerlegbaar door middel van experimenten

Bij het Standaard Model van bewijzen ligt de axiomatische aanname dat experimentele observatie de enige methode is die zinvol bewijsmateriaal oplevert of iets wel of niet bestaat. Deze observatie is alleen zinvol als hij op een wetenschappelijke wijze is verricht. Het resultaat van een experiment wordt alleen als echt geldig beschouwd als het door onafhankelijke partijen kan worden gereproduceerd.[24]

Per definitie, vereist de herhaling van een experiment dat het recursief bepaald wordt - de onafhankelijke wetenschappers kunnen dan dezelfde stappen van het experiment in dezelfde volgorde herhalen, vanaf de zorgvuldig voorbereide beginvoorwaarden, om zo dezelfde resultaten te kunnen produceren. Een concept dat alleen maar zinvol is, als het experiment gezien wordt als een recursieve bewerking op een recursief opgebouwde omgeving.

Aangezien de beperkingen van deze wetenschappelijke methode recursieve reproductie vereist impliceert dit dat men in de natuurkunde gelooft in de volledigheid van recursieve methodes om de natuurkundige werkelijkheid te bepalen. Dit Standaard Model maakt daarom gebruik van een wezenlijke recursieve representatie van de natuur en het is daarom niet verrassend dat fysische hyper-berekenbaarheid zo moeilijk is om aan te tonen: de eigenlijke taal van moderne natuurkunde sluit directe analyse van hyper-berekenbare processen uit.

Hieruit volgt echter niet dat de natuurkunde niets te zeggen heeft over hyper-berekenbaarheid. Het betekent eenvoudigweg dat een dergelijke verklaring zich waarschijnlijk zal baseren op subtiele argumenten en zal waarschijnlijk een andere manier van denken vereisen.

Een hyper-berekenend systeem zou kunnen worden opgebouwd gebruikmakend van een radioactieve steekproef. Aan het gedrag van het systeem ligt de volgende aanname ten grondslag: gegeven een willekeurige radioactieve steekproef, er is een bepaalde tijd waarna een radioactieve stof de helft van haar radioactiviteit heeft verloren (de halveringstijd is de gemiddelde tijd van halvering). Als de radioactieve stof vervalt door α-straling, dan zorgt dit natuurkundig proces ervoor dat het massa verliest aangezien er tijdens dit proces kernen van Helium-atomen worden uitgeworpen. We kunnen het aantal seconden tellen die nodig zijn voordat een bepaalde waarneembare drempel van massaverlies wordt overschreden.

Omdat het verval in het Standaard Model wordt verondersteld als stochastisch, moet het aantal seconden wel willekeurig zijn, dus dit systeem kan beschouwd worden als een "echte willekeurige getallen generator". We bedoelen hiermee een systeem dat gegarandeerd een uitvoer geeft en wat in staat is een positief getal te genereren, maar waarbij het uiteindelijke resultaat niet vooraf kan worden bepaald.

Het is alom bekend dat echte willekeurigheid niet geproduceerd kan worden door een computer. Daarom zou het radioactieve verval de bouw van super-Turing machines moeten toelaten. Echter, de aanname dat de helft van de radioactieve stof uiteindelijk zal vervallen blijkt uiterst subtiel.

Bij het proberen om de aanname uit te drukken komt men tot de conclusie dat er twee radicaal verschillende interpretaties mogelijk zijn en door het vergelijken van de twee interpretaties kunnen we afleiden dat:

- ofwel de aanname is correct en we kunnen een echte oneindige willekeurige getallen generator bouwen, zodat het bestaan van Super Turing en Hypercomputation kan worden bewezen
- of onze aanname is onvolledig: in dat geval kunnen we laten zien dat hyperberekening niet kan worden weerlegd volgens de regels van het Standaard Model (d.w.z. dat de twee concepten logisch gezien onafhankelijk zijn)

Hoewel het radioactief verval een vertrouwd concept is in de moderne natuurkunde, blijkt het een opmerkelijk dubbelzinnige semantiek te hebben. Dit is verontrustend omdat het principe van verval een grote rol speelt bij moderne natuurkunde omdat dit het veelvoorkomende beschrijving van het proton illustreert. Omdat protonen een zeer belangrijk component zijn van elke atoomkern, is het duidelijk dat veel grote samensmeltingstheorien instabiliteit van *alle* atoomsoorten met zich mee zal brengen.

Tegenwoordig zijn honderden elementaire deeltjes bekend, met een gemiddelde levensduur die zich uitstrekt van zeer lang (protonen $> 10^{33}$ jaar), het dagelijkse (vrije neutron ≈ 15 minuten), tot extreem snel ($Z^0 \approx 10^{-25}$ seconden).

Echter, wat bedoelen we precies met "gemiddelde levensduur" in deze context? De standaard definitie is duidelijk, als we bijvoorbeeld starten met een miljoen vrije neutronen, dan verwachten we dat na ruwweg 15 minuten (de halveringstijd van een vrij neutron) ongeveer 500.000 vrije neutronen overhebben en de rest is vervallen. Na nog een kwartier zouden we er nog maar 250.000 overhebben, en na nog een kwartier 125.000. Dit begrip van "halveringstijd" is zo vertrouwd dat het algemeen geaccepteerd is als onbetwiste basis voor experimenten. Bijvoorbeeld een standaard experiment dat gebruikt wordt om het relatieve begrip te ondersteunen dat de tijd voor snel bewegende voorwerpen vertraagt, brengt het meten van de halveringstijd van kosmische stralen met zich mee. Deze snel bewegende deeltjesregen slaat constant neer op Aarde en het is mogelijk om te meten hoe hun aantal in relatie met de hoogte verandert. Hoe lager we onze meting doen, des te langer is de weg dat deze deeltjes hebben afgelegd en verval hebben doorgemaakt, en des te kleiner wordt hun schijnbare stroom. Experimenteel bewijs bevestigt dat de stroom veel langzamer afneemt dan men zou verwachten gegeven de tijd (die gemeten wordt door een waarnemer op Aarde), omdat de snel bewegende deeltjes aanzienlijk minder tijd hebben gevergd om de reis te maken.

Het daadwerkelijk gebruiken van dit concept stelt voor dat natuurkundigen verval niet alleen maar zien als mogelijk maar ook als *noodzakelijk*, maar dit is nergens beschreven in de standaard statistische definities. Veel moderne natuurkundigen zouden verwachten dat ruwweg de helft van een neutronen steekproef vervalt na 15 minuten, maar niemand kan dit garanderen. De mogelijkheid blijft altijd, hoewel haast onwaarschijnlijk, dat géén van de neutronen, zelfs verscheidene dagen later geen verval hebben vertoond. Echter, als dit werkelijk in de praktijk zou gebeuren,

dan is het hoogst onwaarschijnlijk dat een natuurkundige zou zeggen “Aha, dat is moeder natuur die er ons aan laat herinneren dat verval een puur statistisch concept is dat helemaal niet hoeft te gebeuren”. Zij zullen veel eerder zeggen “uw teller is ongetwijfeld stuk”. Dit stelt twee verschillende semantiek voor het verval van onstabiele deeltjes voor. Een is gebaseerd op formele beschrijving, de andere op pragmatisch gebruik.

De rest van deze sectie, die op sommige plaatsen vrij formeel is, is georganiseerd als volgt:

1. We stellen de twee verschillende versies vast van wat het betekent om te zeggen dat "onstabiele deeltjes uiteindelijk vervallen". Ik laat zien hoe deze twee versies kunnen worden uitgedrukt als logische formules, ϕ en ψ.
2. We laten zien dat ϕ en ψ zo gelijkaardig zijn dat er geen experiment bestaat dat onderscheid kan maken tussen hen.
3. We tonen aan dat ϕ een tautologie is van het Standaard Model, en dat Hyper-berekenbaarheid (in feite, Super Turing) een logisch gevold is van ψ
4. Hieruit volgt dat hyper-berekenbaarheid experimenteel onweerlegbaar is in het Standaard Model, omdat:
   - neem aan dat een experiment Hyper-berekenbaarheid weerlegt
   - aangezien Hyper-berekenbaarheid een consequentie is van ψ, is het experimentele systeem een tegenvoorbeeld voor ψ
   - aangezien ϕ en ψ niet door middel van experimenten onderscheiden kan worden, is het experimenteel systeem ook een tegenvoorbeeld voor ϕ
   - aangezien ψ een tautologie is van het Standaard Model, weerlegt het experiment het Standaard Model zelf.

Neem aan dan dat een observatie van een vooralsnog nog niet vervallen onstabiel deeltje p vanaf nu begint, op tijdstip 0. We schrijven $E(p,t)$ om de waarschijnlijkheid aan te tonen dat p nog niet vervallen na verstrijken van tijd t. Het concept van halveringstijd geeft een bijzondere vorm aan het begrip dat $E(p,t)$ naar 0 neigt des te langer we wachten, d.w.z. $E(p,t) \to 0$ als $t \to \infty$. Niettemin kunnen we dit asymptotische gedrag uitdrukken in twee gelijkaardige, maar toch op significant verschillende manieren. In de volgende stellingen, wordt de waarde ϵ altijd verondersteld positief te zijn.

Eerste Interpretatie van $E(p,t) \to 0$ als $t \to \infty$

Deze interpretatie zegt: elk gegeven deeltje p zal of zal niet uiteindelijk vervallen, maar de kans op het niet vervallen zal de 0 naderen met het verstrijken van de tijd. Formeel voldoet p aan:

$$\forall\epsilon.\exists T(\epsilon).\forall t.(t > T(\epsilon)) \to (E(p,t) < \epsilon)$$

d.w.z. dat gegeven een positieve waarschijnlijkheid ϵ, hoe klein dan ook, er een of andere hoeveelheid tijd T zal zijn (de nauwkeurige waarde kan van de keus van ϵ afhangen), dusdanig dat de kans dat p nog steeds bestaat op een gegeven tijdstip t later dan T, lager zal zijn dan ϵ. ■

Tweede Interpretatie van $E(p,t) \to 0$ als $t \to \infty$

Deze interpretatie zegt: elk deeltje p zal uiteindelijk zijn vervallen na een bepaalde tijd $T(p)$, maar we weten niet van te voren welke waarde $T(p)$ zal aannemen. We kunnen er echter vanuit gaan dat als we op vele deeltjes zouden gaan letten, de diverse waarden verdeeld zullen worden zoals verwacht. Formeel voldoet p aan:

$$\forall T(p).\forall t.(t > T(p)) \to (E(p,t) = 0)$$

Dit wil zeggen, dat er met zekerheid een toekomstig tijdstip T bestaat (welke kan variren van deeltje tot deeltje, en welke we misschien niet kunnen berekenen) waarna een deeltje is vervallen. ■

De symmetrie tussen deze twee interpretaties is niet onmiddellijk duidelijk, dus passen we een wiskundige "truc" toe om deze gelijkenis naar voren te brengen. In de tweede interpretatie maken we een botte bewering dat $E(p,t)$ 0 zal zijn met als gevolg dat we de test-parameter "ϵ" niet nodig hebben die in de eerste interpretatie voorkwam. We kunnen ϵ her-introduceren door te observeren dat 0 kan worden gedefinieerd als "het unieke niet-negatieve getal wat lager is dan elk ander positief getal". Hieruit volgt dat:

$$\exists T(p).\forall t.(t > T(p)) \rightarrow (E(p,t) = 0)$$

dezelfde bewering is als

$$\forall \epsilon.\exists T(p).\forall t.(t > T(p)) \rightarrow (E(p,t) < \epsilon)$$

vastgesteld dat $T(p)$ en E onafhankelijk zijn van ϵ en ϵ onafhankelijk is van p. Vervolgens kunnen we de twee interpretaties schrijven in opvallend gelijke termen, waarbij ϵ positief is:

$$\begin{aligned} \alpha(p) \quad &\equiv_{def} \quad \forall \epsilon.\exists T(\epsilon).\forall t(t > T(\epsilon) \rightarrow (E(p,t) < \epsilon) \\ \beta(p) \quad &\equiv_{def} \quad \forall \epsilon.\exists T(p).\forall t(t > T(p) \rightarrow (E(p,t) < \epsilon) \end{aligned}$$

Op deze wijze gezien ligt het onderscheid tussen de twee uitdrukkingen in de manier waarop T gekozen moet worden. In α kiezen we T volgens de beperking ϵ, maar in β gaan we er van uit dat T een kenmerk is van het deeltje zelf. Om het experimentele karakter van deze uitdrukkingen te begrijpen, moeten we E nauwkeuriger bekijken. We hebben aangenomen dat E de asymptotisch nul, exponentieel vervalcurve is van het Standaard Model (zodat de halveringstijd zinvol is). Tijdens enig experiment om de waarheid of onjuistheid van α of β te bepalen, moeten we het deeltje p observeren om te zien of het al dan niet al vervallen is; en door te observeren genereren we nieuwe informatie. In het bijzonder dwingt deze informatie ons om vooraf de aannames met betrekking tot E te wijzigen. Als we, bijvoorbeeld, vinden dat het deeltje inderdaad is vervallen dan is het niet langer zinvol om een positieve waarschijnlijkheid toe te wijzen aan de mogelijkheid dat het 5 seconder later nog niet vervallen zou zijn. Eerder, het moment waarop we weten dat het deeltje is vervallen, alle vooraf bepaalde schattingen van E gaan automatisch naar 0.

Stel dan dat p een deeltje is, en laten we de relatie van p tot α en β beschouwen. Er zijn twee gevallen te onderscheiden, afhankelijk van het feit of p stabiel of onstabiel is. Als we aannemen dat het deeltje in feite stabiel is, verstrekt deze aanname zelf informatie die dwingt dat E zal worden gewijzigd. Voor een stabiel deeltje hebben we, volgens de definitie, $E \equiv 1$, en α en β evalueren beide als *false*. Anderzijds, als, zoals de natuurkundige meer en meer aangeven, p onstabiel is, kunnen we twee subgevallen onderscheiden. Òf er is een tijd t_p waarna het deeltje kan worden gezin als vervallen òf deze tijd is er niet. Als er zo'n tijd bestaat, dan moet E op een bepaald punt worden aangepast zodat $E(p,t) \equiv 0$ for all $t > t_p$. Onder deze omstandigheden zullen zowel α als β evalueren tot *true*. Als zo'n tijd niet bestaat, dan zal er geen enkele aanpassing aan E zich voordoen en kunnen we α en β van elkaar onderscheiden omdat α dan naar *true* evalueert maar β naar *false*.

Om samen te vatten, de enige manier om α van β te kunnen onderscheiden is het zoeken van een deeltje welke schijnbaar onstabiel is, maar welke nooit zal vervallen. Voor strikte personen eindigt dit argument, omdat ze kunnen debatteren (met enige rechtvaardiging) dat een onstabiel deeltje dat nooit vervalt in feite een *stabiel* deeltje is. Echter, als we vanaf het begin het bestaan van onstabiele deeltjes toelaten die nooit zullen vervallen, vereist het bepalen of een deeltje deze eigenschap bezit dat we oneindig lang moeten wachten, welke de wetenschappelijke methode op twee verschillende gronden schend. Ten eerste, een experimenteel resultaat is alleen zinvol als het kan worden herhaald, en een experiment dat oneindig lang doorloopt kan daarna nooit herhaald worden. Ten tweede, moeten de experimenten worden gedaan met uitsluitend "eindige middelen", en dit neemt met zich mee de eis dat ze klaar moeten zijn in eindige tijd. Echter, als we de looptijd van de experimenten beperken dat ze bijvoorbeeld niet langer mogen lopen dan T seconden dan is het duidelijk onmogelijk om nog onderscheid

te maken tussen deeltjes die na T seconden zijn vervallen van de deeltjes die nooit zullen vervallen.

GEVOLG: α en β zijn niet te onderscheiden met behulp van experimenten.

Laten we $Unstable(p)$ schrijven als we bedoelen dat p vanaf het begin een onstabiel deeltje is. We hebben zojuist gezien dat α evalueert tot *true* voor alle schijnbaar onstabiele deeltjes, dus de uitdrukking

$$\phi \equiv_{def} Unstable(p) \rightarrow \alpha(p)$$

is een tautologie (vastgesteld dat men de asymptotische curve van verval accepteert van het Standaard Model); inderdaad, het kan gezien worden als de *definitie* van instabiliteit in het Standaard Model. Omdat α en β niet te onderscheiden zijn door middel van experimenten, kunnen we de uitdrukking

$$\psi \equiv_{def} Unstable(p) \rightarrow \beta(p)$$

zien als een *experimentele tautologie*. Er zouden *logische* tegenvoorbeelden kunnen bestaan tegen ψ maar dit tegenvoorbeeld is experimenteel gezien betekenisloos. Echter, deze uitdrukking ψ is precies de aanname die aan de basis ligt van ons ontwerp van een "echte willekeurige getallen generator", zodat ψ noodzakelijkerwijs Super Turing en Hyper-berekenbaarheid met zich meebrengt.

GEVOLG:ϕ is een tautologie van het Standaard Model, ψ omvat Super Turing, en elke weerlegging van Hyper-berekenbaarheid door middel van experimenten is een weerlegging van het Standaard Model zelf. ■

8 Hypercomputation versus intelligentie

In dit deel van het verslag proberen we de antwoorden te vinden op twee veel gestelde vragen met betrekking tot hypercomputation. De eerste vraag luidt: kunnen mensen hypercomputen? Er bestaan tot nu toe alleen theoretische modellen die hypercomputation mogelijk maken zoals bijvoorbeeld de Oracle Machine en de Accelerated Turing Machine[15]. In het tweede deel proberen we het antwoord te vinden op de vraag of computers kunnen denken.

8.1 Kunnen mensen hypercomputen?

Het zou best wel eens zo kunnen zijn dat wij mensen kunnen hypercomputen. Wiskundigen kunnen redeneren over oneindigheid door bijvoorbeeld te werken met limieten. Tevens zijn mensen in staat om allerlei visuele verwerking te doen. We kunnen bijvoorbeeld objecten herkennen ondanks variatie in positie, grootte en kleur. Ook lijkt het erop dat we het halting probleem kunnen oplossen. Immers als programmeur kun je kijken naar een klein programma en zeggen of het ooit zal stoppen. Maar dat is niet genoeg, als we het halting probleem willen oplossen moeten we van elk programma kunnen bepalen of het ooit zal stoppen of niet. Afgezien van het feit dat men de tijd niet eens heeft om een gigabyte spaghetti code te kunnen lezen, vergeten we tijdens het lezen van de code ook bepaalde details.

8.1.1 Is menselijke kennis onberekenbaar?

Het soort ding dat berekenbaar kan zijn is een functie, dat wil zeggen een reeks van geordende input en output paren dusdanig dat geen twee paren het zelfde eerste element maar verschillende tweede elementen hebben. Ruwweg, een functie is berekenbaar dan en slechts dan als er een algoritme bestaat die hem berekent.

Bringsjord en Penrose beweren dat niet alles van het menselijke redeneren berekenbaar is omdat we onder andere vrije wil bezitten en daarom de capaciteit hebben om random nummers te genereren[28].

Een ander capaciteit waar Bringsjord het over heeft is het redeneren over het oneindige. Als we kunnen redeneren over het oneindige, dan zou dat betekenen dat niet alle menselijke kennis berekenbaar is. Aristoteles maakte in zijn beschouwingen over oneindigheid onderscheid tussen het 'potentieel oneindige' en het 'actueel oneindige'. Met het potentieel oneindige krijg je te maken als je gaat tellen en merkt dat je je telproces nooit tot een einde kunt brengen, bijvoorbeeld bij het tellen van het aantal punten op een lijnstuk. Met het actueel oneindige krijg je te maken wanneer je een oneindige totaliteit in zijn geheel overziet.

Om te kijken of mensen deze capaciteiten hebben, heeft het Rensselaer Polytechnisch Instituut een experiment uitgevoerd. Het doel was om te observeren of zoiets bestaat als vrije wil en redeneren over het oneindige door mensen. In de eerste test werd gekeken of we de capaciteit hebben om random getallen te genereren en in de tweede test werd gekeken of we in staat zijn om oneindigheid te kunnen visualiseren. De testjes werden gedaan door 31 studenten van het Rensselaer Polytechnisch Instituut. De studenten die meededen waren over het algemeen allen eerste jaars informatica studenten[34].

8.1.2 Random getallen generatie testen

In de eerste test wordt aan de studenten gevraagd om een getal tussen de 1.274.862 en 1.972.335 te noteren. De resultaten zijn weergegeven in tabel 1 en 2. Men maakt hierbij gebruik van statistische methoden om te bepalen of de opgegeven getallen echt random zijn. In tabel 1 heeft men alle ingezamelde getallen naast elkaar gezet en gekeken hoeveel een bepaald cijfer in deze reeks voorkomt. Zoals men ziet komt de 0 veel vaker voor dan de 1 en op basis daarvan concludeert men dat de getallen niet echt random zijn. In tabel 2 staan nog meer statistische gegevens, zoals het aantal hoog-laag overgangen en afwisselingen die aantonen dat de opgegeven getallen niet echt random zijn.

Getal	0	1	2	3	4	5	6	7	8	9
Frequentie	19	7	18	15	19	12	13	18	17	12

Table 1: Resultaten van de "random getallen" test (1)

	Verwachte Waarde	**Daadwerkelijke Waarde**
Herhalingen	12	20
Hoog-Laag	50	48
Laag-Hoog	50	52
Afwisselingen	42	43

Table 2: Resultaten van de "random getallen" test (2)

In de tweede test werd gevraagd aan de test populatie om in gedachten twintig keer een munt op te gooien. Voor een munt schreef men een T op en een H voor kop. Men heeft de set van strings opgedeeld in 4 gelijke delen om er makkelijker mee te kunnen rekenen. De resultaten worden weergegeven in tabel 3.

Set 1	Verwacht	Daadwerkelijk	Set 3	Verwacht	Daadwerkelijk
Herhalingen	72	57	**Herhalingen**	72	73
Hoog-Laag	36	48	**Hoog-Laag**	36	41
Laag-Hoog	36	47	**Laag-Hoog**	36	38
Afwisselingen	36	55	**Afwisselingen**	36	34
Set 2	**Verwacht**	**Daadwerkelijk**	**Set 4**	**Verwacht**	**Daadwerkelijk**
Herhalingen	72	58	**Herhalingen**	63	52
Hoog-Laag	36	49	**Hoog-Laag**	32	40
Laag-Hoog	36	45	**Laag-Hoog**	32	41
Afwisselingen	36	53	**Afwisselingen**	32	44

Table 3: Resultaten van de "munt opgooien" test

Statistisch gezien zou er in de eerste set 72 herhalingen moeten zitten, maar het blijken er maar 57 te zijn. Een hoog-laag overgang wordt geteld als een munt wordt opgevolgd door een kop. Wat verder opviel was dat maar liefst 25 van de 31 test personen begonnen met munt. En van de 620 opgegooide munten was het 140 keer kop en 480 keer munt.

8.1.3 Redenatie over oneindige testen

Een test die men heeft bedacht voor het aantonen van de capaciteit om over het oneindige te redeneren is de zogenaamde Achilles Runner test. Deze zit als volgt in elkaar:

Een hardloper rent een halve minuut, houdt dan een pauze van een halve minuut en rent vervolgens een kwart minuut, waarna hij weer een pauze houdt van een kwart minuut, enzovoorts.

De test persoon werd gevraagd hoeveel keer de renner is gestopt en weer verder is gegaan in 2 minuten. Dit representeert een oneindig rekenkundige reeks. 25 studenten gaven het correcte antwoord, 6 gaven een fout antwoord.
En andere test was de Koch Curve, ook wel sneeuwvlok genoemd. Deze Koch Curve heeft bijna iedereen wel eens gezien. Men maakt zon curve door te beginnen met een driehoek te tekenen in een cirkel. Vervolgens voegt men een driehoek toe die 1/3 keer zo groot is al het origineel, op elk van de zijden van de originele driehoek. Men krijgt dan elke iteratie een nieuw figuur zoals te zien is in figuur 6.

Figure 6: Tekenen van een koch curve

De vraag aan de studenten was: Als men deze stappen oneindig zou uitvoeren, wat zou de omtrek dan zijn van het laatste figuur wat je zou tekenen? En zou deze figuur de cirkel vullen?

Het antwoord zou moeten zijn dat de omtrek oneindig is en dat de figuur de cirkel nooit zal vullen. De eerste vraag werd maar door 9 personen goed beantwoord en door 22 personen fout. De tweede vraag werd door 7 mensen correct beantwoord en door 24 incorrect.

Hun uiteindelijke conclusies waren dan ook dat het niet waarschijnlijk is dat mensen echte random getallen kunnen genereren. Misschien dat we in ons brein verfijnde pseudo random algoritmes hebben, maar het is niet duidelijk dat we de capaciteit hebben op echte random getallen te genereren.

Het succes met het redeneren over oneindigheid is in het gunstigste geval inconsistent. Het is niet duidelijk dat de test personen om het even welke capaciteit voor het redeneren over het oneindige hebben gebruikt om conclusies over de bijvoorbeeld convergentie van fractals te maken. De correcte oplossingen kunnen misschien ook aan vroegere kennis of ervaring worden toegeschreven[34].

8.2 Kunnen computers denken?

In dit hoofdstuk gaan we in op de vraag: Kunnen computers denken?. Voor dat je kan zeggen dat een machine kan denken, moet je dit kunnen testen. Hiervoor is de vader van de theoretische computer, Alan Turing, met een idee gekomen van een proefopstelling.

In de Turing Test voert de ondervrager gesprekken via een terminal met twee systemen. Één daarvan is een mens, de andere een machine. Zie figuur 7. De gesprekken kunnen over van alles gaan. Als de ondervrager na een bepaalde tijdsduur van bijvoorbeeld een uur de machine op basis van het gesprek niet kan onderscheiden van een mens, dan zouden we volgens Turing kunnen stellen dat deze machine intelligent is.

Figure 7: De Turing test

Vragen die je zou kunnen stellen zijn bijvoorbeeld: "Wat heb je vannacht gedroomd?" of "Hoe voelt het als je met een hamer op je duim slaat?".

Wel verandert hij hiermee de vraag naar Kan een machine een mens laten denken dat het een mens is? Als dit zo is, dan moet het zo zijn dat deze machine vele capaciteiten heeft van een mens, anders valt hij door de mand als er vragen gesteld worden die dit testen.

Turings voorspelling was dat in het jaar 2000 computers zo intelligent zouden zijn dat ze voor 70% van de tijd niet te onderscheiden zouden zijn van mensen.

Op deze Turing test zijn door meerdere wetenschappers bezwaren geuit. In dit hoofdstuk behandelen we enkele van deze bezwaren en Turings antwoord hierop.

8.2.1 Het "hoofd in het zand" standpunt

Het hoofd in het zand-standpunt stelt dat dit alles te onheilspellend is om over na te denken, en daarom moeten we maar aannemen dat het niet kan. En als we ze kunnen laten denken, dan moeten we dat vooral niet doen: ze zullen dus nooit denken omdat wij dat niet zullen toelaten.

Met dit standpunt kan Turing alleen maar medelijden mee hebben[9].

8.2.2 Chinese kamer argument

John Searle ontwikkelde in 1980 een "gedachten experiment" dat (volgens Searle[35]) bewijst dat een computer nooit echt kan denken en dat de Turing Test niet voldoende is om te bewijzen dat een computer kan denken. Searle is er van overtuigd dat een dergelijk systeem de Turing Test kon doorstaan. Het systeem, gevisualiseerd in figuur 8, zit als volgt in elkaar:

Stel er bevind zich in een kamer een persoon die wel Engels begrijpt maar geen Chinees. Tevens bevindt zich in de kamer een complex boek met regels die aangeven welk chinees symbool naar buiten moeten worden gestuurd als een bepaald chinees symbool binnenkomt. Deze Chinese kamer noemen we dan systeem M. Searle beweert dat het systeem M slaagt voor de Turing Test voor het begrijpen van Chinees (de persoon die buiten de kamer staat denkt dat hij met een Chinees aan het praten is), maar dat het systeem M in werkelijkheid geen Chinees begrijpt. Het vervangt gewoon de symbolen aan de hand van die regels in het boek.

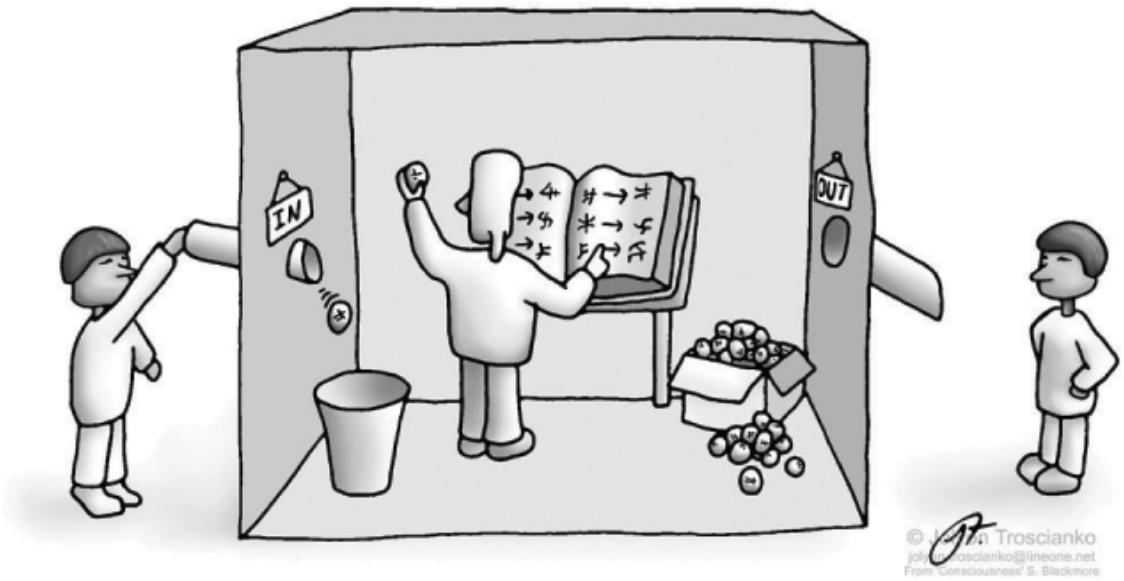

Figure 8: Het chinese kamer experiment

Er zijn meerdere wetenschappers die kritiek hebben op het bewijs van Searle. Een daarvan is Peter Kugel. Die zegt dat de Chinese Kamer geen bewijs kan zijn waarom computers niet kunnen denken omdat Searle beweerd dat zijn Chinese Room een simulatie is van wat een computer allemaal kan (namelijk domweg regels toepassen), maar in werkelijkheid kan een computer veel meer.

Als we het boek, dat zich in die Chinese kamer bevind, zichzelf zouden laten schrijven dan kan het bepaalde dingen herinneren en het gedrag aanpassen aan de ervaringen dat het heeft opgedaan[31]. Dit zou vervolgens intentionaliteit bereiken dat precies nodig is om computers te laten begrijpen.

8.2.3 Theologisch bezwaar

Het theologische bezwaar tegen machines die bewustzijn bereiken is het eerste bezwaar dat Turing voorstelt in zijn paper, dit bezwaar komt waarschijnlijk voort uit de (christelijke) godsdienst: alleen mensen hebben een ziel hebben gekregen van God, geen dier of machine heeft een ziel en zal daarom nooit kunnen denken. Met andere woorden, intelligentie vereist een ziel.

Turing verwerpt dit argument in zijn paper en als tegenargument gebruikt hij dat dit slechts vanuit een Christelijk standpunt geld en niet in andere godsdiensten. In de Boeddhistische cultuur heerst er namelijk de overtuiging dat de menselijke zielen overgebracht kunnen worden naar dierlijke organismen. Turing gaat echter niet in welk standpunt correct is, er zijn daarover al teveel oorlogen bestreden[9].

8.2.4 Mathematisch bezwaar

Een bezwaar dat werd geuit door J.R. Lucas en P. Kugel baseert zich op Gödels incompleetheidstheorie. Gödels theorie toont aan dat elk consistent en krachtig formeel systeem beperkingen heeft, zelfs als hij een oneindige capaciteit zou hebben[33][32].

Turings benadering daarin is dat mensen ook beperkingen hebben en fouten maken. Misschien dat in de toekomst er machines zullen zijn die intelligent genoeg zijn om te concurreren met mensen.

Een ander tegenargument is dat Lucas niet bewijst dat zulke beperkingen ook niet van toepassing zijn op het intellect van mensen. Gödels theorie heeft alleen betrekking op consistente formele systemen, wij mensen zijn misschien onbeperkte formele systemen die fouten maken[9].

8.2.5 Andere bezwaren m.b.t. ontbrekende eigenschappen

Turing noemt dit 'Arguments from Various Disabilities'. De bezwaren richting zich op de stelling dat machines kunnen niet uit emotionele redenen handelen. Wanneer zij handelen kunnen ze niet voelen en er zijn geen emotionele gevolgen.

Andere eigenschappen die een machine nooit zou kunnen bezitten, welke vaak naar voren worden gebracht, zijn: "vriendelijk, vindingrijk, mooi, vriendschappelijk zijn, initiatief hebben, humor hebben, onderscheid kunnen maken tussen slecht en goed, fouten maken, verliefd worden, van aardbeien genieten en dergelijke.

De Benadering van Turing: wij kunnen niet weten hoe een machine voelt aangezien wij geen machines zijn. Machines zijn beperkt omdat ze meestal maar een zeer kleine capaciteit hebben. Ook vind hij deze argumenten nogal bevooroordeeld. Niet alleen omdat men er maar van uit gaat dat dergelijke eigenschappen in computers niet voor kunnen komen, ook omdat men verwacht dat deze eigenschappen in een machine moeten voor komen om aan de standaards van intelligentie te voldoen. Een van deze niet-voorkomende eigenschappen is het maken van fouten. Daarbij wordt een mechanische fout en een denkfout nogal eens door elkaar gehaald. Een los draadje kom je tegenwoordig nauwelijks tegen, en inderdaad, een denkfout maakt een computer eigenlijk per definitie niet. Maar als het gaat om het nabootsen van mensen is dit weer geen probleem, dan zorgen we ervoor dat ze zo nu en dan fouten maken. Ook wordt gezegd dat computers niet over zichzelf kunnen denken. Maar het ziet er naar uit dat computers binnenkort met programma's uitgerust worden die hun eigen resultaten controleren en zo in staat zijn hun programmas aan te passen[9].

8.2.6 Lady Lovelaces bezwaar

Lady Lovelaces bezwaar is dat computers (in haar tijd) niet creatief kunnen zijn. Om creatief te zijn moet je iets nieuws kunnen uitvinden. Maar computers vinden niets uit, ze voeren alleen programmas uit die wij ze geven[30][29].

Het antwoord van Turing hierop is, dat we eigenlijk alleen maar een klein stukje hoeven te maken: de mogelijkheid om te kunnen leren. Daarna zou de machine, zoals een pasgeboren kind, alles from scratch" kunnen leren.

Een variant op haar stelling is dat computers ons niet zouden kunnen verrassen.

Het antwoord van Turing daarop is dat computers ons toch kunnen verrassen. Omdat wij nog wel eens fouten maken kan het best zijn dat een programma een verrassend resultaat geeft[9].

8.2.7 Continuteit met het zenuwstelsel

Het zenuwstelsel is zeker geen discrete machine en lijkt dus totaal niet op een digitale computer die zich alleen in discrete toestanden bevinden. Dit maakt ze berekenbaar, want als in het programma van een discrete machine staat dat de leeskop een positie naar links geschoven moet worden dan weet je ook dat de leeskop daar gaat staan. In een continue machine gaat dit niet. Staat daar een operatie die de leeskop $\sqrt{2}$ positie-eenheden laat opschuiven, dan is de nieuwe plaats van de kop niet meer zo makkelijk te bepalen en wordt de machine onnavolgbaar. Bij het zenuwstelsel kan een kleine fout in de informatie over de grootte van een zenuwprikkel die een neuron benvloedt, een enorm verschil uitmaken aan de grootte van de uitgaande impuls. Men kan debatteren dat het

niet waarschijnlijk is dat het gedrag van het zenuwstelsel met een discrete machine na te bootsen is.

Turings commentaar op dit bezwaar was dat dit feit geen enkele invloed heeft op het slagen van de Turing Test. De ondervrager heeft namelijk geen voordeel bij het verschil van een discrete-state en een continu systeem.

8.2.8 Serendipiteits argument

Een ander argument is het zogenaamde "serendipity" argument. We geven een Finite State Machine (FSM) dat willekeurige Engelse zinnen kan genereren de naam P. Tijdens een Turing Test kan P geluk hebben en de ondervrager op het verkeerde been zetten. Echter geeft Selmer toe in zijn paper dat het vrijwel onmogelijk is dat P de ondervrager op het verkeerde been zet — maar het is wel mogelijk[27].

Anderen zeggen dat de vraagstelling van de Turing Test te zwak is. Een computer hoeft maar uitgerust te worden met diverse programmas die ervoor zorgen dat zijn antwoorden zeer menselijk lijken en hij zal door de test komen. Tussenpozen tussen vraag en antwoord, random antwoorden hier en daar, zelfs als de ondervrager vraagt wat voor haar hij heeft, kan hij dit even bij elkaar liegen.

9 Conclusies

Hypercomputation strekt zich uit tot vele vakgebieden: filosofie, informatica en natuurkunde en er wordt al sinds de oudheid over gediscussieerd. De eerste discussies op basis van bewezen theoriën zijn ontstaan naar aanleiding van de uitvinding van Alan Turing's machine (De Turing Machine). Hypercomputation biedt de mogelijkheden om "onoplosbare" problemen op te lossen en spreekt daarom erg tot de verbeelding. Ondanks alle tegenargumenten gelooft het overgrote deel van de wetenschap minstens in het bestaan van Hypercomputation. De vraag echter blijft of (en vooral hoe) we het kunnen gebruiken. Zowel de theoretische als de praktische ideeën hierover zijn zeer divers en bevinden zich nog allemaal in de ontwikkelingsfase.

References

[1] Turing's Ideas and Models of Computation, Eugene Eberbach, Dina Goldin, Peter Wegner, 2004

[2] Introduction to the Theory of Computation, M. Sipser, 1997.

[3] Alan Turing and the Development of the Electronic Computer, Jonathan Salinas, Found at: http://www.math.rutgers.edu/courses/436/Honors02/turing.html

[4] Languages and Machines, Thomas A. Sudkamp, 1997

[5] On computable numbers, with an application to the Entscheidigungsproblem, A.M. Turing, 1936

[6] Abstraction to Implementation, Robert M. Keller, Found at:http://www.cs.hmc.edu/claremont/keller/webBook/

[7] Church-Turing Thesis, Found at http://en.wikipedia.org/wiki/Church-Turing_thesis

[8] A Lecture to the London Math. Society on 20'th February 1947, in A.M. Turing's Ace Report of 1946 and Other Papers, Alan Turing, 1986

[9] Computing Machinery and Intelligence, Alan Turing, 1950,

[10] Hilbert's Tenth Problem, Found at:http://logic.pdmi.ras.ru/Hilbert10/stat/stat-eng.html

[11] Natural Computation and Non-Turing Models of Computation, B.J. Maclennan, 2003

[12] Transcending Turing Computability, B.J. Maclennan, 2003

[13] The physical and philosophical implications of the Church-Turing Thesis, Eleni Pagani, 2004

[14] http://plato.stanford.edu/entries/spacetime-supertasks/, Jon Pérez Laraudogoitia, 2004,

[15] Hypercomputation: computing more than the Turing machine, Toby Ord, 2002

[16] Hypercomputation: philosophical issues, B. Jack Copeland, 2003

[17] Quantum Algorithm for Hilberts Tenth Problem, Tien D.Kieu, 2004

[18] Quantum Hypercomputation, Tien D. Kieu, 2002

[19] Hypercomputation with quantum adiabatic processes, Tien D. Kieu, 2003

[20] Physical Hypercomputation and the ChurchTuring Thesis, Oron Shagrir and Itamar Pitowsky, 2003

[21] Hypercomputation, Gert-Jan C. Lokhorst, 2001

[22] Non-Turing Computers and Non-Turing Computability, Mark Hogarth, 2001

[23] Toward a Formal Philosophy of Hypercomputation, Selmer Bringsjord and Michael Zenzen, 2002

[24] Computation and Hypercomputation, Mike Stannett, 2003

[25] Three aspects of super-recursive algorithms and hypercomputation or finding black swans, Mark Burgin, Allen Klinger, 2003

[26] Super-Turing Computation: A Case Study Analysis, Keith Douglas, 2003

[27] Could, how could we tell if, and should androids have inner lives?, Selmer Bringsjord, 1994.

[28] The modal argument for hypercomputing minds, Selmer Bringsjord, Konstantine Arkoudas, 2003

[29] Can a Computing Machine Be Genuinely Creative, Selmer Bringsjord, Found at: http://www.rpi.edu/ brings/select.html

[30] Creativity, the Turing Test, and the (Better) Lovelace Test, Selmer Bringsjord and Paul Bello, 2000, Found at: http://www.rpi.edu/ faheyj2/SB/SELPAP/DARTMOUTH/lt3.pdf.

[31] Can Computers Be Genuinely Intelligent?, Peter Kugel, Computer Science Department, Boston College, 2001, found at: http://www.cs.bc.edu/ kugel/Publications/Hyper.pdf

[32] Computing Machines Cant Be Intelligent (... and Turing Said So), Peter Kugel, 2002

[33] Minds, Machines and Gdel, JR Lucas, 1961, Found at: http://users.ox.ac.uk/ jrlucas/Godel/mmg.html

[34] Do Humans Hyper-Compute? A Study of Human Capacity for Infinitary Reasoning, Mitch Mailman, 2003.

[35] Minds, Brains, and Programs: Behavioral and Brain Sciences 3, John Searle, 1980