Aanzet tot een formele beschrijving van leerprocessen

Jeroen van Maanen*

31 oktober 2001

Samenvatting
Als leren neerkomt op het vergaren van kennis en vaardigheden en als het mogelijk zou zijn om kennis en vaardigheden te meten zoals massa of kracht, dan zou het mogelijk moeten zijn om een formele theorie over leerprocessen op te stellen. In dit artikel wordt een meetmethode voor kennis beschreven. Mijn these is dat alle kennis en ervaring die binnen ons bereik ligt ook binnen het bereik van de meetmethode ligt. Naast deze meetmethode bevat dit artikel ook een voorstel voor de structuur van een formele leertheorie die op de meetmethode gebaseerd is.

1 Introductie

Meten is weten

Wat is het dat een appel naar beneden doet vallen? Het lijkt vanzelfsprekend, maar als je er over nadenkt ligt de werking van de zwaartekracht helemaal niet zó voor de hand. Newton had er de bewegingen van de planeten voor nodig en genoeg fansatie om een appel te beschouwen als een klein maantje met een langgerekte baan die erg snel de moederplaneet tegenkomt. Toen hij dat verband gevonden had kon hij bewegingen van heel verschillende objecten binnen één begrippenkader beschrijven en voorspellen. Hij had de systematiek van de aantrekkingskracht tussen objecten gevonden. Daarmee heeft hij onze kennis over het verschijnsel zwaartekracht vergroot. Voorwaarde voor het beschrijven van systematiek is een beschrijving van feiten die het mogelijk maakt om verschijnselen te meten. Newton introduceerde bijvoorbeeld het begrip massa en leende het begrip euclidische afstand en de bijbehorende beschrijvingsmethode gebaseerd op coördinaten om objecten en bewegingen te beschrijven. Deze begrippen vormen een meetmethode voor de bewegingen van objecten.

Stel dat we voor het begrip leren iets vergelijkbaars willen doen. We hebben dan begrippen nodig die goed te meten zijn zodat we de systematiek van gemeten verschijnselen kunnen bestuderen. In dit artikel is het uitgangspunt dat we het begrip kennis kunnen verhelderen zoals Descartes het begrip afstand verhelderd heeft. We gaan dus op zoek naar een meetmethode voor kennis.

Zo op het eerste gezicht lijkt een zoektocht naar een meetmethode voor kennis een hopeloze onderneming. Het lijkt op het zoeken naar een meetmethode voor *lift*, de kracht die vleugels doet stijgen. Vóór de gebroeders Wright was de algemene stemming dat het principieel uitgesloten was dat mensen konden vliegen. Vogels kunnen nu eenmaal vliegen,

* E-mail: `Jeroen.van.Maanen@xs4all.nl`

maar andere dieren of dingen niet. *Lift* is een intrinsieke eigenschap van vogels die we kunnen constateren, maar waar we verder geen vat op hebben behalve door eigenschappen van vogels te beschrijven. Met het begrip kennis lijkt iets soortgelijks aan de hand te zijn in die zin dat kennis uitsluitend in hersenen voorkomt, dat het niet mogelijk lijkt om kennis los te zien van de fysieke vorm waarin we kennis tot nu toe altijd gezien hebben. Ik[1] denk dat ik de filosoof John Searle goed parafraseer met de uitspraak "hersenen hebben mentale toestanden, andere dingen niet" [Sea80]. Als kennis gelijk staat met iets weten en weten is een mentale toestand, dan impliceert deze uitspraak dat er naast eigenschappen van hersenen niets over kennis te zeggen is. Laten we er voorlopig vanuit gaan dat het mogelijk is dat kennis net als aerodynamische eigenschappen en vrije val bewegingen los van de fysieke vorm gemeten kan worden en dat ze een systematiek vertoont die in onze eigen bouwsels zou kunnen worden toegepast.

Waar moeten we die systematiek in zoeken? Laten we het met introspectie proberen. Waar zijn we in dit artikel mee bezig? We proberen iets over leren te weten te komen. We proberen er iets over te... leren. Wat zoeken we? Systematiek. Wat doen we als we die gevonden hebben? Dan verwerken we die systematiek in ons verwachtingspatroon. Met andere woorden: we passen ons model van de wereld aan. Dus als we ons model van de wereld aanpassen naar aanleiding van onze ervaringen en het aangepaste model beschrijft meer systematiek dan het oude model, dan zeggen we dat we iets geleerd hebben. Alexander van den Bosch stelt in zijn proefschrift [Bos01] dat het met opzet toepassen van drogredenen een goede strategie is bij het zoeken naar nieuwe hypothesen. Daar maak ik in dit geval dankbaar gebruik van door de zojuist geformuleerde causale relatie om te draaien en te gebruiken als *definitie* van leren. In dit artikel zal dus de volgende definitie gehanteerd worden. Een systeem met een intern model van de wereld *leert* als het systeem het model aanpast op basis van de ervaringen met de wereld en als het zich zó gedraagt dat de in het model beschreven systematiek toeneemt met het toenemen van de ervaringen. Hierbij moet worden aangetekend dat de zinsnede "in het model beschreven systematiek" opgevat kan worden als een definitie voor de subjectieve *kennis* van het systeem over de wereld. Ik zal beargumenteren dat deze definitie, mits creatief toegepast, alle situaties dekt waarin geleerd wordt. Nu komt het er op aan om aan te tonen dat de termen in deze definities voldoende meetbaar zijn om als basis voor een formele leertheorie te dienen. Hiervoor zullen we verderop in het artikel de termen systematiek en model verder uitdiepen.

2 Doelstelling

Het doel heiligt de middelen

Het doel van dit artikel is uiteenzetten hoe het mogelijk zou zijn om lerend gedrag in zijn algemeenheid formeel te *beschrijven* en daarmee te meten (hier ligt wel de afzwakking "in principe" op de loer). Ter ondersteuning van deze doelstelling zal ik betogen dat de vorm van deze beschrijving de mogelijkheid biedt om op termijn machines te construeren en te programmeren die binnen de voorgestelde beschijvingsmethode vallen en waarvan het gedrag

1. Ik schrijf in de eerste persoon als ik, zoals in dit geval, mijn subjectieve interpretatie geef of als ik een term een betekenis geef die afwijkt van de gebruikelijke betekenis uit het woordenboek of het vakgebied.

in mijn ogen ook als lerend zal moeten worden aangemerkt. Ik zal niet proberen een ontologische theorie over de *aard* van leren te presenteren. Ik voel me niet ter zake kundig om uitspraken over de aard of de essentie van kennis, begrip, het bewustzijn of van wat dan ook te doen en ik zal me daar dan ook verre van houden.

De uitdaging in de doelstelling zit in de qualificatie formeel. We zullen van zoveel mogelijk aspecten van lerend gedrag proberen een beschrijving in psychologische termen te vertalen naar een beschrijving in termen van machinaal reproduceerbare processen. Daarbij zullen de resulterende methodes om het leerkarakter van processen te beschrijven de vorm moeten aannemen van een formeel model. In een boek over Romeinse stedenbouw [Mac81] word het woord 'formeel' gebruikt als de naam van een houten staketsel waarmee een stenen boogconstructie tijdens de bouw wordt gesteund. Als de sluitsteen van de boog geplaatst is kan het formeel verwijderd worden. De boog blijft dan opzichzelf staan. Dat komt overeen met wat ik met een formeel model voor het beschrijven van lerend gedrag voor ogen heb: een constructie die voorkomt dat een theorie al tijdens het opstellen ervan in elkaar valt. Als de theorie af is zullen de gedefineerde begrippen ook zonder formeel model hun samenhang moeten kunnen behouden en zal de theorie moeten kunnen fungeren als een verklaring van het gedrag van lerende organismen en systemen. De theorie zal ook als leidraad moeten kunnen dienen bij het construeren van nieuwe lerende systemen.

3 Benaderingen

De waarheid ligt in het midden

We kunnen tot een beschrijving van lerend gedrag komen door de methodes van een aantal vakgebieden toe te passen. We zullen eerst globaal inventariseren hoe lerend gedrag binnen de psychologie beschreven wordt. Vervolgens gaan we te rade bij de kansrekening die ons de mogelijkheid geeft om modellen te construeren waarin zeer uiteenlopende verbanden tussen waarnemingen vervat zijn. De klasse van modellen die op deze manier geconstrueerd kunnen worden is groot genoeg om de mogelijkheid open te houden dat zelfs het complete verwachtingspatroon van een mens uitputtend te beschrijven is met een model uit deze klasse. Deze modellen zijn specifiek genoeg om ze wiskundig te kunnen analyseren. De informatica geeft ons een formele definitie van machinaal gedrag. Ik zal, zonder daarbij tot de technische details af te dalen, schetsen op welke manieren binnen de informatica gedrag beschreven wordt. Daarna bekijken we een wiskundige theorie over de mate van willekeur in een reeks symbolen die aangeduid wordt met Kolmogorov complexiteit. Ook hier zal ik weer afzien van een technische verhandeling. Met behulp van Kolmogorov complexiteit kunnen we een getal toekennen aan de hoeveelheid regelmaat die in een model vervat is. Deze maat kunnen we gebruiken om te meten hoeveel kennis een lerend systeem over de wereld verzameld heeft. In de slotparagraaf zal ik schetsen hoe, met het volgens deze benaderingen verkregen materiaal, een formeel model voor lerend gedrag eruit zou kunnen zien.

3.1 Psychologie

Kennis is macht

Leren is een vorm van gedrag en valt om die reden binnen het terrein van de psychologie voor zover het gedrag van mensen en dieren betreft. De nadruk op *gedrag* betekent niet dat het om een *behavioristisch* model voor leren zou gaan. Met gedrag bedoel ik alle wisselwerking tussen een systeem en de omgeving. Daarbij is het niet noodzakelijk dat het systeem waarvan we het gedrag bestuderen die interactie ook controleert. Ik hanteer dus de natuurkundige opvatting van gedrag (zoals in "het gedrag van een gas bij gelijk volume onder een toenemende temperatuur"). In een psychologische context betekent dit dat ik catatonisch gedrag of het anderszins ondergaan van stimuli zonder daarop te reageren ook als gedrag beschouw. De parabool die een steen beschrijft als je er een schop tegen geeft is bijvoorbeeld net zozeer gedrag als de reactie van een hond die in dezelfde situatie een stuk uit je broek hapt [WBJ67]. Dit maakt het mogelijk om een zeer ruime keuze aan potentieel lerende systemen en interacties tussen deze systemen en hun omgeving te bestuderen zonder condities in aanmerking te nemen die in mijn ogen niets met leren te maken hebben. Tegenover deze zeer ruime definitie van gedrag staat dat we slechts een zeer beperkt aspect van gedrag bekijken. Het gaat om de invloed van het gedrag op de hoeveelheid kennis over de omgeving — de wereld — .

Een algemene definitie van leren in psychologische termen is: *leren is het aanpassen van gedrag op basis van ervaring ter bevrediging van interne behoeften.* Hierbij moet worden aangemerkt dat voor het bevredigen van behoeften vaak energie, geld, vrijheid van handelen of andere voorwaarden nodig zijn. Het creëren van deze voorwaarden wordt vaak een afgeleide behoefte, een doel op zichzelf. Een van de belangrijkste voorwaarden voor effectief gedrag, in termen van bovenstaande interne behoeften, is kennis over de omgeving. Vandaar dat we ons zullen concentreren op het verzamelen van kennis als hoofddoel van een lerend systeem. Andere criteria als het verzamelen van voedsel en het zoeken of construeren van onderdak kunnen worden gezien als reacties op beperkingen in de interactie tussen lerend systeem en omgeving. Als het organisme sterft daalt de hoeveelheid kennis over de omgeving in het systeem in korte tijd naar nul, dus overlevingsgericht gedrag kan verklaard worden als gedrag tot behoud van kennis.

Ook het aanleren van vaardigheden kan worden gezien als het vergroten van kennis over de omgeving. Bij het aanleren van vaardigheden wordt kennis opgedaan over de interactie tussen het individu en de omgeving. Het individu weet na het aanleren van de vaardigheid wat het moet doen om een bepaald doel te bereiken. Hierbij kan het interessant zijn om te kijken waar de kennis wordt 'opgeslagen'. Met name vaardigheden leren we vaak niet met onze hersenen maar met ons hele lichaam. Voor onze theorie is dat geen probleem. *Hoe* de kennis binnen het lerende systeem gerepresenteerd is staat los van de *hoeveelheid* kennis die binnen het systeem gerepresenteerd is.

Ik zou hier de hypothese willen opperen dat alle drijfveren uiteindelijk te herleiden zijn tot behoefte aan macht over de omgeving. Dit betekent dat het systeem invloed op de omgeving kan uitoefenen door uit verschillende strategieën te kiezen (dat lijkt mij ook de enig zinvolle definitie van vrijheid). Strategieën kunnen alleen beoordeeld worden op basis van kennis over de omgeving. Bijvoorbeeld als iemand als Kafka niet weet dat een deur open zou kunnen gaan door er aan te trekken in plaats van er tegen te duwen, kan hij deze strategieën niet goed beoordelen. Mogelijkerwijs zijn dus alle (psychologische) drijfveren te herleiden tot de behoefte aan kennis over de omgeving.

We zullen ook proberen om de termen waarin we de beschrijving van lerend gedrag zullen gieten zó algemeen te houden, dat deze beschrijving zowel van toepassing is op het gedrag van individuen, als op het gedrag van groepen, organisaties of zelfs de gehele wetenschap.

3.2 Kansrekening

Het kan vriezen, het kan dooien

Het is binnen de Artificiële Intelligentie gebruikelijk om de kennis van een systeem over de omgeving te beschrijven in termen van predicaten. Het systeem redeneert over deze predicaten met behulp van de klassieke predicatenlogica, eventueel uitgebreid met modale operatoren of andere constructies die onvolledige kennis en/of gedeeltelijk incorrecte feiten toestaan. Deze aanpak brengt echter de nodige problemen met zich mee. Aangezien leren inhoudt dat de kennis van een systeem over de omgeving toeneemt met de tijd wil dat zeggen dat, zolang het systeem nog leert, de kennis over de omgeving onvolledig is. Een strategie die mensen hanteren om daarmee om te gaan is om waar nodig aannamen te doen en die te herzien zodra meer gegevens bekend zijn. Het vinden van aannamen die logisch consistent zijn met de huidige kennis is buitengewoon moeilijk. Veel op predicatenlogica gebaseerde systemen zijn daarom, of erg langzaam, of gebruiken heuristieken voor het omzeilen van inconsistenties, die vaak al te eenvoudig tot absurde afleidingen leiden. Een ander probleem is dat de logische implicatie incompatibel is met de causaliteitsrelatie. Het blijkt zeer moeilijk te zijn om causaliteit goed te formaliseren binnen de predicatenlogica [Grü98].

Een mogelijke uitweg uit deze problematiek is het gebruik van kansrekening in plaats van predicatenlogica. De kansrekening is van meet af aan opgezet om te redeneren over waarnemingen die onzeker zijn omdat ze nog gedaan moeten worden en waarnemingen waarvan onzeker is hoe ze tot stand zijn gekomen. Verder staat de kansrekening toe dat verschillende aannamen die elkaar onderling uitsluiten naast elkaar bestaan. Het is bijvoorbeeld denkbaar dat iemand in een casino met zichzelf afspreekt: als ik nu win speel ik met het gewonnen kapitaal verder, als ik verlies ga ik naar huis. Aan de mogelijke uitkomsten van de waarneming (winst, verlies) worden waarden toegekend die maatgevend zijn voor de gemiddelde verwachtte uitkomst bij herhaling van het experiment.

De overstap van predicatenlogica naar kansrekening is niet zo groot als het op het eerste gezicht misschien lijkt. De axiomatisering van de kansrekening lijkt erg op die van de predicatenlogica. De predicatenlogica kan zelfs worden opgevat als een limietgeval van de kansrekening (*Cf.* Popper [Pop68, §48] die Wittgenstein's *Tractatus Logico-Philosophico* [Wit18, prop. 5.152] citeert).

Een probleem van de predicatenrekening dat niet direct door het gebruik van kansrekening wordt opgelost is het vraagstuk van de elementaire feiten. Het is al een oud vraagstuk binnen de filosofie hoe het mogelijk is om op grond van een noodzakelijk beperkte hoeveelheid waarnemingen iets met *zekerheid* vast te kunnen stellen. In de kansrekening vertaalt dit probleem zich naar het vraagstuk hoe het mogelijk is om op grond van een beperkte hoeveelheid waarnemingen de kans op een verschijnsel vast te stellen.

Hume stelde, al vóór de formalisering van predicatenlogica en kansrekening, vast dat dit probleem, het inductieprobleem, in zijn sterkste formulering onoplosbaar is. Het feit dat de zon sinds mensenheugenis elke dag weer opgekomen is, geeft ons niet de garantie dat hij morgen wéér opkomt. Hume constateerde dat het niet mogelijk is om uit een verzameling waarnemingen, stellige uitspraken af te leiden over gebeurtenissen die (nog) niet zijn waargenomen. Hij ging zelfs zover te stellen dat kennis over toekomstige gebeurtenissen slechts *a priori* kennis, dus onafhankelijk van de waarnemingen kan zijn.

Binnen de predicatenlogica kunnen waarnemingen inderdaad hoogstens gebruikt worden voor het bepalen van de waarheid/onwaarheid van elementaire proposities voor zover ze betrekking hebben op die waarnemingen. Als een model binnen de predicatenlogica al kennis representeert over gebeurtenissen die (nog) niet zijn waargenomen, moet die kennis volgen uit axioma's, dat wil zeggen *a priori* gegeven formules.

Voordat we de mogelijkheden van de kansrekening op dit punt kunnen overzien, moeten we eerst afspreken wat het getal dat we de kans op een verschijnsel noemen betekent. De discussie over de betekenis van het woord 'kans' is nog in volle gang [LOG95]. Pogingen om 'kans' op te vatten als een objectieve meetbare eigenschap van een reeks waarnemingen [MIS39, POP68] zijn tot op heden gestrand op de constatering dat extra informatie de kans op een verschijnsel kan beïnvloeden [RIS89]. Bijvoorbeeld als we over satelietfoto's van het weer over de hele aardbol beschikken kunnen we de kans dat het morgen zal regenen in Amsterdam beter voorspellen (de kans ligt dichter bij 1 of 0), dan wanneer we het zonder die informatie moeten doen. Daarom zullen we ons richten op de vraag: hoe zeker kunnen we van een geobserveerde regelmatigheid zijn, gegeven een bepaalde hoeveelheid waarnemingen?

Ik zal proberen aan te geven dat het mogelijk is hierop een antwoord te geven als we afzien van de eis dat dit oordeel objectief moet zijn. Daarmee bedoel ik dat er vele mogelijkheden zijn om op basis van dezelfde waarnemingen kansen toe te kennen, die welliswaar van elkaar kunnen verschillen, maar waarvan de toegekende kansen naarmate de lijst waarnemingen langer wordt, dichter bij elkaar komen te liggen. Hiermee blijft de mogelijkheid open dat het net zo min mogelijk is om objectief de kans vast te stellen dat het morgen regent, als om te beslissen of alle raven zwart zijn. Hume's constatering dat je het met elke afzonderlijke bewering oneens kunt zijn zonder onredelijk te worden blijft onaangetast.

Aan de andere kant is het wèl mogelijk om een referentiekader te kiezen en binnen dat kader subjectieve kansen toe te kennen op zo'n manier dat individuen die verschillende referentiekaders hanteren het op de lange duur altijd eens worden over een gegeven bewering. De mogelijkheid dat er een reeks beweringen is zodat twee individuen het altijd oneens zijn over één of andere bewering uit die reeks is niet uitgesloten.

Turings theorie van machinaal gedrag heeft een stelling opgeleverd die impliceert dat er universele referentiekaders zijn. Het redeneren op basis van kansen die zijn toegekend op basis van een universeel referentiekader zal op de lange termijn tot dezelfde conclusies leiden als redeneren op basis van kansen toegekend binnen het kader van een willekeurig ander universeel referentiekader. Een systeem dat kansen schat binnen een universeel referentiekader kan worden gezien als een systeem dat zich in elk ander referentiekader kan 'inleven', inclusief elk ander universeel referentiekader[2]. Voordat we dieper ingaan op de fundering van de kansrekening als methode om onvolledige kennis te modelleren zal de formele beschrijving van machinaal gedrag nader uitgewerkt worden.

2. Het kenmerkend vermogen van een bewust schepsel om zich in te leven in een ander schepsel wordt uitgebreid aan de orde gesteld in *The Mind's I* [DH81, Hst. 24].

3.3 Machinaal gedrag

Verstand op nul, blik op oneindig

Al in de dertiger jaren formuleerde Alan Turing een model voor het beschrijven van machinaal gedrag: de Turing-machine [Tur36]. Het belang van dit model is vervat in de zogenaamde Church-Turing these. Die stelt dat alle berekenbare regelmatigheden in gedrag nagebootst kunnen worden met een Turing-machine. Dit model is om twee redenen relevant voor het formaliseren van lerend gedrag.

Ten eerste volgt uit de Church-Turing these dat een beschrijving van lerend gedrag gegoten moet kunnen worden in de vorm van een algoritme, dat wil zeggen een programma voor een Turing-machine. Dat wil zeggen dat dit programma in principe uitgevoerd zou moeten kunnen worden door een gewone computer. Of een computer dat een leer-algoritme uitvoert ook leert is een vraag waaraan ik, zoals aangekondigd in de inleiding, in dit artikel geen aandacht zal besteden.

Ten tweede houdt de Church-Turing these een formele definitie van regelmaat in die, met enige moeite, omgezet kan worden in een kwantitatieve definitie van regelmaat. Het resultaat is een complexiteitsmaat die gebruikt kan worden om een bruikbare schatting van kansen op elementaire gebeurtenissen te genereren. De stap van Turing's model van machinaal gedrag naar het schatten van kansen is niet triviaal en zal worden uitgewerkt in paragraaf 3.4.

Laten we er van uit gaan dat elk mechanisch systeem vertaald kan worden in een symbolen verwerkend systeem. Het is bijvoorbeeld mogelijk om de begintoestand van bijvoorbeeld een stoommachine te beschrijven in termen van pijpdiameters, coördinaten, materiaaleigenschappen enzovoort. Vervolgens kan de configuratie van de machine op een later tijdstip berekend worden met behulp van rekenregels die afgeleid zijn van de natuurwetten. Een dergelijke berekening simuleert het gedrag van de stoommachine. Dergelijke simulaties zijn inmiddels dagelijkse kost voor ingenieurs. Bij het berekenen van latere toestanden van de machine wordt uitsluitend gewerkt met de gegeven beschrijving van de begintoestand en de rekenregels.

Bij het beschrijven van symboolverwerkende systemen maakte Turing gebruik van een formeel model. Hij beperkte zich tot systemen van een betrekkelijk eenvoudige vorm, de zogenaamde Turing-machines. De werkwijze van Turing was om naast de begintoestand van de machine ook de rekenregels volledig uit te schrijven. De werking van een Turing-machine wordt volledig bepaald door een eindige beschrijving van die machine zoals beschreven in appendix A.

De reikweidte van deze constructie wordt onderstreept door de Church-Turing these die stelt dat alle berekenbaar gedrag exact gereproduceerd kan worden met een Turing-machine. Hoewel deze stelling niet wiskundig bewezen kan worden heeft ze binnen de informatica de status van een natuurwet. Het vertrouwen van informatici in deze stelling heeft zelfs zoveel indruk gemaakt dat een vooraanstaand fysicus als Richard Feynman de hypothese geopperd heeft dat het gedrag van het hele fysische universum tot op quantummechanisch niveau *exact* te beschrijven is met een universele machine [Fey82]. (Feynman gaat uit van een ander, maar equivalent, model van machinaal gedrag dan Turing.)

De grote ontdekking die Turing deed is dat het mogelijk is om een Turing-machine te definiëren die alle Turing-machines kan simuleren. De reden daarvoor is dat er een Turing-machine T_U denkbaar is die gegeven een beschrijving van een (andere) Turing-machine T_x en de invoer voor T_x, uitrekent welke uitvoer T_x geproduceerd zou hebben bij de gegeven invoer. De codering van de regels van een Turing-machine voor gebruik met een universele Turing-machine heet een *programma* voor die universele Turing-machine. Een universele Turing-machine die als invoer een dergelijk programma en een codering van de oorspronkelijke invoer krijgt, simuleert het gedrag van de oorspronkelijke machine. Turing bewees ook dat er oneindig veel universele Turing-machines zijn, meestal met verschillende coderingen voor de regels. De meeste programmeertalen zoals we die nu kennen zoals Pascal, C, Lisp en Prolog zijn theoretisch te gebruiken als coderingssytemen voor universele Turing-machines. De universaliteit van deze talen blijkt uit het feit dat er vertalers zijn tussen deze talen. Het is bijvoorbeeld mogelijk om een Pascal programma naar een C programma te vertalen zonder enig verlies aan betekenis. Het resulterende C programma doet precies hetzelfde als het origineel. De meeste machinale vertalers vertalen programma's die zijn gescheven in een abstacte programmeertaal naar een taal die meer lijkt op de lijst regels waarmee we Turing-machines gedefiniëerd hebben en die met machinetaal aangeduid wordt. Machinetaal kan direct door een digitale computer uitgevoerd worden.

Later willen we het behalve over berekeningen ook over machinale processen hebben. Met een machinaal proces bedoelen we een, niet noodzakelijk eindige, reeks gebeurtenissen waarvan elke gebeurtenis op een berekenbare manier volgt uit de voorafgaande. Hiertoe zullen we gebruik maken van het feit dat er Turing machines zijn die een onbegrensde reeks invoersymbolen transformeren tot een potentiëel onbegrensde reeks uitvoersymbolen. In appendix A wordt beschreven op welke manier de standaard Turing-machine moet worden aangepast om dit soort processen te beschrijven. In wiskundige terminologie heet de standaard Turing-machine ook wel een discrete Turing-machine. De aangepaste variant voor onbegrensde reeksen heet continu. De reden hiervoor is dat de invoer van een discrete Turing-machine eindig te beschrijven is en, als de machine stopt, de uitvoer ook. In tegestelling daarmee is de uiteindelijke invoer van een continue Turing-machine een oneindige rij waarop de machine continu kan doorrekenen. Het resultaat kan dan ook weer een oneidige rij symbolen zijn. De verzameling van alle oneindige rijen over een bepaald alfabet is in wiskundige zin een continuüm. Een continue Turing-machine wordt ook wel aangeduid met de term monotone Turing-machine omdat de uitvoer monotoon toeneemt met de invoer. Net als universele discrete Turing-machines zijn er ook universele continue Turing-machines.

Als we een continue Turing-machine gebruiken om het gedrag van een systeem of organisme te modelleren, coderen we de waarnemingen van het systeem en schrijven bieden die aan als invoer voor de continue Turing-machine. Vervolgens laten we de machine werken tot hij probeert meer symbolen te lezen dan we aangeboden hebben. De symbolen die in de tussentijd door de machine als uitvoer geproduceerd zijn vatten we op als beschrijvingen van acties. Vervolgens bepalen we welke waarnemingen volgen op de beschreven acties. De beschrijving van deze waarnemingen bieden we aan als extra invoer en we laten de machine verder rekenen.

Bij wijze van gedachtenexperiment kan ook de hele wetenschap opgevat worden als lerend systeem. Als invoer bieden we dan alle waarnemingen van alle experimenten uit de historie tot nu toe aan. Als de machine een goede representatie is van het wetenschapsbedrijf, produceert het alle experimenten die er gedaan zijn als uitvoer. (Toegegeven, dit is een onwaarschijnlijke aanname, tenzij de werkelijkheid waarin we leven uitputtend beschreven kan worden als een algoritme voor een Turing-machine. Een aanname die falsifieerbaar is, maar nog niet gefalsifieerd [FEY82].)

Als we een continue Turing-machine gebruiken om een theorie over de omgeving van een lerend systeem te modelleren, dan zetten we de machine bij wijze van spreken in z'n achteruit. We coderen de waarnemingen van het systeem zoals die zouden kunnen worden geproduceerd als uitvoer door de Turing-machine. We kijken dan of er een reeks symbolen is zodat als we die als invoer aanbieden, de machine exact de beschrijving van de ervaringen reproduceert. Als de beschrijving van de ervaringen eindig is, mag de continue Turing-machine op de gegeven invoer meer uitvoersymbolen genereren. Zolang de beschrijving van de ervaringen maar een beginstuk is van de uitvoer. De aangeboden invoer kan dan opgevat worden als de codering van de randvoorwaarden en eventuele correcties.

We zullen in de volgende paragraaf zien dat deze opzet de mogelijkheid biedt om op een verstandige manier kansen toe te kennen aan theorieën en toekomstige waarnemingen.

3.4 Willekeur

When in doubt: flip a coin

In deze sectie zullen we een formele definitie van de hoeveelheid willekeur[3] in een beschrijving bekijken. Met deze maatstaf kunnen we ook de mate van regelmaat in een beschrijving vaststellen. Dit gaat op dezelfde manier als het wegen van sigarenrook. Weeg eerst de sigaar. Rook hem vervolgens op, daarbij de weegschaal als asbak gebruikend. Doe de peuk bij de as en trek van het oorspronkelijke gewicht van de sigaar het gewicht van de overblijfselen van af.

Voor het schatten van kansen is willekeur een centraal begrip. Iets dat zeker is krijgt kans 1, iets dat onmogelijk is krijgt kans 0. Hoe zekerder iets is, hoe dichter bij 1 de geschatte kans zal liggen en hoe onwaarschijnlijker iets is, hoe dichter de geschatte kans bij 0 zal liggen. De afstand van de geschatte kans tot de absolute grenzen 1 en 0 wordt bepaald door de mate van willekeur in het verschijnsel. Elke regelmaat in het optreden van een verschijnsel leidt tot een betere voorspelling van dat verschijnsel en trekt dus de geschatte kans naar één van beide uitersten toe.

Om kwantitatieve uitspraken te kunnen doen over kansen op verschijnselen op grond van een beperkte hoeveelheid waarnemingen, stellen we ons voor dat we een beschrijving van alle waarnemingen van het systeem tot op een bepaald moment tot onze beschikking hebben. Een *theorie* die deze waarnemingen verklaart kan dan gegoten worden in de vorm van een algoritme, een programma voor een universele Turing-machine, dat de beschrijving van de waarnemingen exact reproduceert. In het geval van een theorie die de waarnemingen niet exact voorspelt, kan het algoritme bestaan uit een sub-algoritme dat de voorspelling in het ideale geval uitrekent, aangevuld met aanpassingen die verwerkt worden in het resultaat van het sub-algoritme. Samen moeten deze delen alsnog exact de beschrijving van de waarnemingen reproduceren. Een triviale toepasing hiervan is een theorie die helemaal niets voorspelt. In dat geval bevat de lijst met uitzonderingen een exacte beschrijving van de waarnemingen. Het zal duidelijk zijn dat in dat geval de lengte van een beschrijving van het sub-algoritme dat deze simpele theorie plus de lengte van een beschrijving van de lijst met uitzonderingen ongeveer gelijk is aan de lengte van de beschrijving van de waarnemingen die we proberen te reproduceren. Aan de andere kant als we een niet triviale theorie hebben, bijvoorbeeld de theorie van Newton over de bewegingen van hemellichamen, en we proberen een lijst met waargenomen planeetstanden te reproduceren, dan kunnen we volstaan met een beschrijving van het sub-algoritme (dat hangt niet af van het aantal waarnemingen en kunnen we negeren

3. Ik gebruik hier 'willekeur' in een vrij nauwe betekenis, als vertaling van het Engelse 'randomness'.

als het om erg veel waarnemingen gaat) plus een lijst met data en planeetnamen en kleine correcties. De feitelijke planeetstanden hoeven we niet letterlijk in het algoritme op te nemen, want die kunnen uit de overige gegevens gegenereerd worden met behulp van het sub-algoritme dat Newton's mechanica beschrijft. Dat wil dus zeggen dat het algoritme dus significant korter zal zijn dan de oorspronkelijke letterlijke beschrijving van de waarnemingen.

Als we de oorspronkelijke beschrijving en het algoritme beide in het binaire alfabet opschrijven (dat wil zeggen dat we ons tot twee symbolen beperken), dan kunnen we het verschil van de lengte van de oorspronkelijke beschrijving en de lengte van het kortste algoritme dat deze beschrijving exact reproduceert opvatten als de mate van regelmaat die de beschrijving bevat. De lengte van het korste algoritme noemen we de Kolmogorov complexiteit van de (beschrijving van de) waarnemingen [LV93]. Als er geen enkel algoritme is dat de beschrijving van de waarnemingen letterlijk reproduceert en dat echt korter is dan de oorspronkelijke beschrijving, dan zeggen we dat de waarnemingen volstrekt willekeurig zijn; we kunnen er geen regelmaat in ontdekken. Dit oordeel hangt overigens af van de specifieke universele Turing-machine waarmee we de algoritmen interpreteren. Een reeks waarnemingen kan volstrekt willekeurig zijn in termen van een universele Turing-machine maar toch regelmatig zijn in termen van een andere. Het is wel zo dat als een oneindige rij symbolen oneindig veel beginstukken heeft die willekeurig zijn in termen van een universele Turing-machine, dan heeft die rij oneindig veel willekeurige beginstukken in termen van elke andere universele Turing-machine.

Voor een continue universele Turing-machine kunnen we nu de kans bekijken dat als we kop-of-munt gooien voor de invoersymbolen, de uitvoer een (eventueel verlengde) exacte reproductie van de beschrijving van de waarnemingen is. Deze kans ligt dicht bij twee tot de macht min de Kolmogorov complexiteit van de beschrijving van de waarnemingen in termen van de gegeven continue Turing-machine. Deze kansmaat kan met behulp van Bayesiaanse statistiek gebruikt worden voor het maken van inductieve afleidingen. Om deze reden heeft Ray Somonoff onafhankelijk van Kolmogorov dezelfde complexiteitsmaat gedefiniëerd [SOL64]. Aangezien Gregory Chaitin [CHA87] ook weer onafhankelijk van Kolmogorov en Solomonoff dezelfde definitie opstelde, is de volle naam van deze complexiteitsmaat: Kolmogorov-Solomonoff-Chaitin complexiteit.

Onder de aanname van de Church-Turing these kunnen we nu een willekeurige universele continue Turing-machine zien als een theorie voor het beschrijven van waarnemingen. De kans op een specifieke waarneming a gegeven een historie d is dan de conditionele kans die volgens de standaard formule uit de kansrekening gelijk is aan $\mathbf{P}(a \mid d) = \mathbf{P}(d\,a)/\mathbf{P}(d)$. Waarbij $\mathbf{P}$ de eerder beschreven kansmaat is. Merk op dat volstrekt tegenstrijdige waarnemingen een positieve kans kunnen hebben, zolang de som van de kansen maar gelijk is aan 1.

De zwakke plek van deze opzet is niet zozeer dat de keuze van de universele machine arbitrair is. Dat heeft nauwelijks effect op de geschatte kansen op de lange termijn. De moeilijkheid is dat de Kolmogorov complexiteit voor alle behalve een eindig aantal beschrijvingen onberekenbaar is. Het is principieel onmogelijk om het kortste algoritme te berekenen dat een bepaalde uitvoer oplevert. Een voor de hand liggende methode om dit probleem te omzeilen is om te eisen dat het begin van de invoer een algoritme beschrijft dat binnen een analyseerbare klasse van algoritmen valt. Een dergelijke klasse noemen we dan een *hypotheseklasse*. Bij computer-programma's die ontwikkeld zijn om cognitief gedrag te modelleren, zoals BACON, MYCIN en SOAR, is een dergelijke hypotheseklasse expliciet of impliciet gegeven.

De kans op een reeks waarnemingen gegeven een hypotheseklasse kan met behulp van de kansrekening eenvoudig bepaald worden. Als de hypotheseklasse overzichtelijk genoeg is kan de kans zelfs uitgerekend worden. Inductieve inferentie is dan de taak om het algoritme uit de hypotheseklasse te vinden waarvoor de resulterende beschrijving van de waarnemingen zo kort mogelijk is. Als er dan meerdere mogelijkheden zijn verdient het aanbeveling om meer waarnemingen te doen. Deze methode om kansen in te schatten op basis van een overzichtelijke hypotheseklasse is ontwikkeld door Jorma Rissanen [Ris78, Ris89] en onafhankelijk van Rissanen door de informatie-theoretici Wallace en Boulton [WB68]. Het onderliggende principe om de lengte van de codering van een beschrijving van de waarnemingen zo kort mogelijk te maken wordt in het Engels aangeduid als het *Minimum Description Length Principle* (MDL).

De allermooiste variant zou zijn om de hypotheseklasse langzaam ingewikkelder te laten worden. Langzaam genoeg om de rekentijd overzichtelijk te houden en snel genoeg om in de limiet dezelfde kansen aan waarnemingen toe te kennen als de universele Turing-machine. De haalbaarheid van deze mogelijkheid is voor zover ik weet nog onderwerp van studie.

4 Machinaal Leren

Wat de theorie had moeten doen, leert de praktijk

Op basis van de voorgaande paragraaf zal ik nu schetsen hoe een formele beschrijving van lerend gedrag er uit zou kunnen zien.

In de paragraaf over willekeur lag de nadruk op het beschrijven van waarnemingen. Echter een zeer klein deel van de mensheid legt zich geheel toe op het beschrijven van waarnemingen. En zelfs dat deel publiceert deze beschrijvingen in plaats van deze voor zichzelf te houden. Een belangrijk deel van een lerende systemen is belast met het genereren van uitvoer. Leren valt nog niet mee als werkelijke interactie met de omgeving niet mogelijk is.

De moeilijkheid bij het analyseren van acties van een lerend systeem is het te hanteren criterium. In onderzoek binnen de Artificiële Intelligentie is het gehanteerde criterium vaak extern. Hoe vaak geeft een medisch expertsysteem een correcte diagnose? Is de ontleedboom van een natuurlijke taal herkenner correct? Aan de andere kant is het criterium voor mensen die leren uiteindelijk intern. Mensen vinden bepaalde omstandigheden prettiger dan andere en proberen zich zo lang en zo vaak mogelijk prettig te voelen. Het lijkt niet mogelijk om het begrip 'prettig' in een zelfde formeel jasje te duwen als we met het begrip 'kans' gedaan hebben. Aan de andere kant kunnen we wel zeggen dat voor een lerend systeem 'leren' belangrijk is. In paragraaf 3.1 heb ik beargumenteerd dat leren uiteindelijk neerkomt op het vergroten van de hoeveelheid kennis over de omgeving. In paragrafen 3.2 en 3.3 heb ik laten zien dat een intern model van een lerend systeem kan worden opgevat als een machinaal berekenbare kansverdeling over (exacte beschrijvingen van) alle mogelijke reeksen waarnemingen. Aangezien een dergelijke kansverdeling berekenbaar moet zijn is er, gegeven een specifieke universele Turing-machine, een programma dat die kansverdeling uitrekent. Dit programma heeft een lengte die uitgedrukt kan worden in bits. Onder bepaalde voorwaarden kunnen we deze lengte zien als de hoeveelheid kennis in het model. We kunnen nu zeggen dat een systeem leert in de mate waarin de kennis in het interne model toeneemt met de tijd.

Dit is een subjectief ordeel in de zeer strikte zin dat het afhangt van de gekozen universele Turing-machine die we gebruiken om de kennis in het model mee te meten en de manier waarop we het model en de waarnemingen coderen.

De voorwaarde waaronder de lengte van het programma dat een model codeert gezien mag worden als een maat voor de hoeveelheid kennis in dat model is dat het model geen regelmatigheden mag beschrijven die zich niet in de waarnemingen van het systeem hebben voorgedaan. Dit wordt afgedwongen door het MDL-principe dat is behandeld in paragraaf 3.4. Een leeralgoritme moet een model kiezen zodat de lengte van het programma dat het model codeert plus de lengte van de invoer reeks die nodig is om een exacte beschrijving van de waarnemingen te produceren minimaal is.

Een systeem kan worden gezien als een lerend systeem als er een manier bestaat om een intern model en de interacties van het systeem met de omgeving zo te coderen dat het aan bovenstaande voorwaarden voldoet. De leerprestatie van het systeem is dan de mate waarin de acties van het systeem er in slagen om mogelijke regelmatigheden die nog niet in de waarnemingen uit het verleden besloten liggen bloot te leggen. Als een dergelijk algoritme gevonden zou worden voor een langzaam in complexiteit toenemende hypotheseklasse zoals beschreven aan het eind van paragraaf 3.4, dan zou dat algoritme met recht een universeel leer-algoritme mogen heten.

In dit model is het dus niet zo dat de interne processen van een lerend systeem in de voorgaande zin altijd tot uitdrukking komt in het gedrag, de uitvoer, van het systeem. Het is heel goed mogelijk dat de omgeving van dien aard is dat zelfs een wiskundig optimaal lerend systeem niet genoeg systematiek kan ontdekken om een verandering van gedrag te rechtvaardigen. Bijvoorbeeld, als de reeks invoersymbolen vanaf een zeker moment constant wordt, dan zal het optimale gedrag bestaan uit het volkomen willekeurig genereren van acties. Dit gedrag zal pas veranderen zodra de reeks invoersymbolen een andere regelmaat gaat vertonen, die correlaties vertoont met de gegenereerde uitvoer. Het voorgestelde model van lerende systemen is dus geen *behavioristisch* model. Het feit dat een systeem zich aanpast aan een gecompliceerde en veranderende omgeving maakt het tot een lerend systeem. Let wel, een omgeving die willekeurige waarnemingen genereert is net zo min een gecompliceerde omgeving als een omgeving die steeds dezelfde waarnemingen genereert.

Het toetsen van dit model in psychologische zin aan lerende organismen lijkt vooralsnog moeilijk. Daarvoor is een theorie nodig die de stimuli die van invloed zouden kunnen zijn op het gedrag van een organisme vertaalt in een overzichtelijke gegevensstroom. Gezien het feit dat zowel Kolmogorov complexiteit en het MDL principe uitgaan van het *exact* reproduceren van deze stroom ligt het evenaren van het menselijk leervermogen op een dergelijke stroom gegevens nog ver buiten bereik. Mogelijkerwijs kunnen binnen sterk gereduceerde omgevingen zinvolle vergelijkingen tussen werkelijk lerende organismen en lerende algoritmen gemaakt worden.

Het voorgestelde model gaat er vanuit dat kennis verzamelen over de omgeving de basis vormt van lerend gedrag. Onder de aanname dat de Church-Turing these correct is kan alle kennis over de omgeving gegoten worden in de vorm van een algoritme dat de informatie over de omgeving reproduceert gegeven een beschrijving van randvoorwaarden en correcties. Het evaluatie criterium voor een lerend systeem is dan de effectiviteit van het algoritme dat de kennis representeert. Hoe korter het algoritme plus de beschrijving van de randvoorwaarden en correcties is, hoe meer regelmaat de waarnemingen blijken te vertonen. De acties van het lerende systeem kunnen vervolgens zo gekozen worden dat de verwachting van de toename van de lengte van het algoritme plus be beschrijving van de randvoorwaarden en correcties gemaximaliseerd wordt.

Toegepast op het wetenschapsbedrijf betekent dat dus dat wetenschappers blijven proberen theorieën op te stellen waarmee de werkelijkheid zo efficiënt mogelijk beschreven kan worden. Naast het formuleren van geheel nieuwe theorieën kan dat door een bestaande theorie bondiger te formuleren, door bestaande theorieën te integreren, door een theorie te ontwikkelen die randvoorwaarden genereert uit andere randvoorwaarden zodat ze niet meer gespecificeerd hoeven te worden. Aangezien wetenschappers verschillende referentiekaders hanteren zullen ze het vaak niet eens zijn over welke theorie het allerbeste is. Dat betekent dat er experimenten gedaan moeten worden die data produceren op grond waarvan ze het alsnog eens worden. Verder kan er nog speculatieve wetenschap bedreven worden in die zin dat experimenten bedacht kunnen worden waardoor die theoriën die nu niet efficiënt zijn, het op grond van de nieuwe resultaten toch kunnen worden. Een belangrijk punt bij het vergelijken van twee theorieën is wel dat *eerst* overeenstemming bereikt wordt over 1) de te reproduceren reeks gegevens en 2) het onderliggende begrippenkader waarin beide theorieën geformuleerd (desnoods vertaald) moeten worden. Anders zijn paradoxen niet te vermijden.

Het lijkt dus mogelijk om een willekeurig lerend systeem te zien als een systeem met een intern model van de omgeving als we er vanuit gaan dat we de interactie van het systeem met de omgeving exact kunnen beschrijven en als we het interne model kunnen coderen (decoderen is hier misschien een beter woord) in de vorm van een berekenbare kansverdeling over beschrijvingen van mogelijke reeksen waarnemingen. Met wat specialistisch gereedschap zoals een universele continue Turing-machine, Kologorov complexiteit en het *Minimum Description Length Priciple* kunnen we de voorwaarden vastleggen waaronder we durven te beweren dat het model de kennis van het systeem over de omgeving op basis van de gedane waarnemingen representeert. Het inherent subjectieve karakter van de meetmethode en de voorwaarde dat alle waarnemingen van het systeem exact beschreven moeten worden, maken het vooralsnog moeilijk om deze formele beschrijving voor bestaande lerende systemen te gebruiken. De beschrijving is wel uitermate geschikt om als basis te dienen voor nieuw te construeren lerende systemen.

A Turing machines

Regels zijn regels

Een Turing-machine [Tur36] kan worden beschreven door een lijst te maken met regels van de vorm: als in toestand p, het symbool a zichtbaar is, schrijf dan symbool b, doe een stap naar links of rechts en neem toestand q aan. Deze regels moeten worden uitgevoerd op een in vakjes verdeelde strook papier vaarvan steeds slechts één vakje zichtbaar is. Elk vakje kan precies één symbool bevatten. We eisen dat de lijst uit slechts eindig veel regels bestaat en dus maar eindig veel toestanden beschrijft. Een van deze toestanden is de begintoestand.

Op de strook papier kan aan het begin van de berekening een reeks symbolen worden geschreven. Deze reeks heet de *invoer* van de berekening. We zorgen ervoor dat het eerste lege vakje vakje na de invoer zichtbaar is. We kunnen dan herhaaldelijk de regel die van toepassing is uitvoeren. We nemen aan dat er altijd hoogstens één regel van toepassing is. Op het moment dat voor de nieuwe toestand van het systeem en het op dat moment zichtbare symbool geen enkele regel van de lijst van toepassing is stopt de berekening. Als de strook papier gedurende de berekening te kort dreigt te zijn, moet er een stuk aangeplakt worden. De reeks symbolen die op de strook papier achterblijft als de machine stopt is de *uitvoer* van de berekening.

Het is evident dat het niet uitmaakt of de regels door een mens of door een automaat worden uitgevoerd. De uitvoer wordt volledig bepaald door de invoer, de lijst regels en de begintoestand. We noemen de lijst regels samen met de specificatie van de begintoestand een Turing-machine. Aan het eind van deze appendix zijn enige voorbeelden van berekeningen met Turing-machines gegeven.

Om processen te beschrijven gebruiken we Turing-machines met drie stroken papier. Op de bovenste strook kunnen tijdens het proces invoersymbolen geschreven worden. De middelste strook dient als kladpapier en wordt op dezelfde manier gebruikt als de enige strook van een eenvoudige Turing-machine. Op de onderste strook kan de machine tijdens het proces uitvoersymbolen schrijven. Welke regel van toepassing is, hangt af van de symbolen die zichtbaar zijn op de bovenste en de middelste strook. Op de bovenste en de onderste strook kan alleen naar rechts bewogen worden of pas op de plaats gemaakt. Naar links bewegen op deze stroken is uitgeloten. Aangezien de bovenste strook nu de invoer bevat, kunnen we de middelste en de onderste strook leeg veronderstellen aan het begin van het proces. Als we een eindig aantal symbolen op de invoerstrook schrijven kunnen we regels toepassen tot de machine een symbool probeert te lezen dat we nog niet geschreven hebben. De uitvoer die tot dan toe geschreven is kan gezien worden als het begin van de uitvoer van het proces op een oneidige rij symbolen waarvan we alleen een beginstuk hebben behandeld. We kunnen meer symbolen op de bovenste strook schrijven en de machine verder laten rekenen met als resultaat de uitvoer van het proces op het verlengde beginstuk.

Net als universele discrete Turing-machines zijn er ook universele continue Turing-machines. De enige extra procedure die een universele continue Turing-machine moet uitvoeren is het kopieëren van het programma van de invoerstrook naar de middelste strook.

Ophogen

1

$\texttt{\# 1 1 1 1 1}\boxed{\texttt{\#}}$

a

2

$\texttt{\# 1 1 1 1 1 1}\boxed{\texttt{\#}}$

Werking

n	q	l	q'	s	r
a	*1*	#	*2*	1	>

Optellen

1

$\texttt{\# 1 1 \# 1 1}\boxed{\texttt{\#}}$

a

2

$\texttt{\# 1 1 \# 1}\boxed{\texttt{1}}\texttt{\#}$

b

3

$\texttt{\# 1 1 \#}\boxed{\texttt{1}}\texttt{\# \#}$

d

3

$\texttt{\# 1 1}\boxed{\texttt{\#}}\texttt{1 \# \#}$

c

4

$\texttt{\# 1 1 1}\boxed{\texttt{1}}\texttt{\# \#}$

e

4

$\texttt{\# 1 1 1 1}\boxed{\texttt{\#}}\texttt{\#}$

Werking

n	q	l	q'	s	r
a	*1*	#	*2*	#	<
b	*2*	1	*3*	1	<
c	*3*	#	*4*	1	>
d	*3*	1	*3*	1	<
e	*4*	1	*4*	1	>

Suggestie

Probeer volgens deze regels nog een paar andere optellingen te maken. Wat gebeurt er als één van beide getallen nul is? Merk op dat de lees-/schrijfkop in het begin op de goede plaats moet staan.

Universele Turing-machine

start

$\texttt{\$ \$ }\textit{1}\texttt{ \# }\textit{2}\texttt{ 1 > \$ \$ \# 1 1}\boxed{\texttt{\#}}$

…

stop

$\texttt{\# 1 1 1}\boxed{\texttt{\#}}\texttt{\# \#} \cdots$

maar ook

start

$\texttt{\$ \$ }\textit{1}\texttt{ \# }\textit{2}\texttt{ \# < \$ }\textit{2}\texttt{ 1 }\textit{3}\texttt{ 1 < \$ } \cdots \texttt{ \$ }\textit{4}\texttt{ 1 }\textit{4}\texttt{ 1 > \$ \$ \# 1 \# 1}\boxed{\texttt{\#}}$

…

stop

$\texttt{\# 1 1}\boxed{\texttt{\#}}\texttt{\# \#} \cdots$

Werking

De werking van een universele Turing-machine is te complex om hier in tabelvorm uit te schrijven. Zie bijvoorbeeld [LP81] voor een complete beschrijving.

In de tabellen staat in kolom n de naam van de regel, in kolom q de toestand van de machine, in kolom l het gelezen symbool, in kolom q' de toestand van de machine na het toepassen van de regel, in kolom s het te schrijven symbool en in kolom r de richting van de lees-/schrijfkop.

Het idee van een universele Turing-machine U is dat de complete beschrijving P_M van een willekeurige Turing-machine M samen met (een beschrijving van) de invoer x als invoer aan U wordt aangeboden. Voor geschikte machines U geldt dat U bij invoer $P_M\,x$ altijd hetzelfde resultaat oplevert als M bij invoer x.

Referenties

[Bos01] Alexander P.M. van den Bosch. *Rationality in Discovery: A Study of Logic, Cognition, Computation and Neuropharmacology*. Institute for Logic, Language and Computation, 2001.

[Cha87] Gregory J. Chaitin. *Algorithmic Information Theory*. Cambridge University Press, 1987.

[DH81] Daniel C. Dennet and Douglas R. Hofstädter, editors. *The Mind's I*. Basic Books, 1981.

[Fey82] Richard P. Feynman. Simulating physics with computers. *International Journal of Theoretical Physics*, 21(6/7):467–488, 1982.

[Grü98] Peter D. Grünwald. *The Minimum Description Length Principle and Reasoning under Uncertainty*. Institute for Logic, Language and Computation, 1998.

[Log95] James Logue. *Projective Probability*. Oxford philosophical monographs. Clarendon Press, Oxford, 1995.

[LP81] Harry R. Lewis and Christos H. Papadimitriou. *Elements of the Theory of Computation*. Prentice Hall Inc., Englewood Cliffs, NJ, 1981.

[LV93] Ming Li and Paul M.B. Vitányi. *An Introduction to Kolmogorov Complexity and its Applications*. Springer Verlag, 1993.

[Mac81] David Macaulay. *De Stad: Het verhaal van de Romeinse stedebouw*. Ploegsma, Amsterdam, 1981.

[Mis39] Richard von Mises. *Probability, Statistics and Truth*. William Hodge & Co., London, 1939.

[Pop68] Karl Popper. *The Logic of Scientific Discovery*. Harper Torch Books, New York, NY, 1968.

[Ris78] Jorma J. Rissanen. Modeling by shortest data description. *Automatica*, 14:465–471, 1978. Journal of the IFAC.

[Ris89] Jorma J. Rissanen. *Stochastic Complexity in Statistical Enquiry*. World Scientific, Singapore, 1989.

[Sea80] John R. Searle. Minds, brains, and programs. *The Behavioral and Brain Sciences*, 3, 1980.

[Sol64] Ray J. Solomonoff. A formal theory of inductive inference, part I. *Information and Control*, 7:1–22, 1964.

[Tur36] Alan M. Turing. On computable numbers, with an application to the entscheidungsproblem. In *Proceedings of the London Mathematical Society, Series 2*, volume 42, pages 230–265. London Mathematical Society, 1936.

[WB68] C.S. Wallace and D.M. Boulton. An information measure for classification. *Computer Journal*, 11:185–195, 1968.

[WBJ67] Paul Watzlawick, Janet Beavin Bavellas, and Don D. Jackson. *Pragmatics of Human Communication: A Study of Interactional Patterns, Pathologies and Paradoxes*. Norton, New York, NY, 1967.

[Wit18] Ludwig Wittgenstein. *Tractatus Logico-Philosophicus*. ?, 1918.