3

De connectionistische mode[1]

W.J.M. Levelt

Directeur Max Planck Instituut voor Psycholinguïstiek, Nijmegen

Verreweg het meest boeiende aspect van een halve eeuw informatisering is mijns inziens het ontstaan van een nieuw computationeel model van de menselijke geest. Dat model heeft zijn wortels in een aantal meer of minder samenhangende ontwikkelingen en ontdekkingen.

Wellicht de belangrijkste daarvan is Alan Turings formalisering van het begrip 'berekening', *computation*. Een functie is (effectief) berekenbaar wanneer de waarde van die functie langs zuiver mechanische weg, dat wil zeggen zonder tussenkomst van een interpreterende instantie, gevonden kan worden.

Het klassieke computationele model

Het spreekt nu al een halve eeuw tot de verbeelding dat wellicht elk goed definieerbaar proces, of het nu een bewijsvoering is, een logische inferentie, het ontleden van een Engelse of Nederlandse zin, het oplossen van een schaakprobleem, het vinden van een driedimensionale interpretatie voor een tweedimensionaal patroon, zou kunnen bestaan uit elementaire, mechanisch uitvoerbare berekeningsstapjes. Die stapjes zijn dan van zuiver syntactische aard, dat wil zeggen dat ze uitsluitend betrekking hebben op de vorm, niet op de inhoud van de ingevoerde expressie.

Turing verschafte ook het model van de logische machine die zulke berekeningen kan uitvoeren. Het is een combinatie van een eindige automaat en een oneindig lange tape. Op een apart stukje van de tape is een eindig aantal elementaire operaties opgeslagen die de automaat kan uitvoeren (dat is het programma); de rest van de tape is er voor de invoer, voor de opslag van tussenresultaten en voor de uitvoer. Programma en data zijn van dezelfde aard, maar worden bij de uitvoering van een berekening strikt gescheiden gehouden. Tenslotte is er dan Turings bewijs dat er universele Turingmachines bestaan, machines die de werking van elke andere Turingmachine kunnen simuleren. Dat betekent dat Turings notie van berekenbaarheid, van een effectieve procedure, van volstrekt algemene aard is.

Het moet worden opgemerkt dat deze eerste stap op de weg naar informatisering werd gezet zonder dat er computers waren. Turings definitie is volmaakt onafhankelijk van de implementatie die men kiest voor de mechanische procedure. Het doet er niet toe of dat een VAX is of menselijke hersenen of een con-

nectionistisch netwerk. De essentie van alle berekening ligt in het *stored program concept:* de scheiding van een beperkt aantal syntactische operaties en een onbeperkte dataverzameling waarop die operaties kunnen aangrijpen; semantiek en implementatie zijn irrelevant. Zolang die essentie is gerealiseerd in de *virtuele* machine, dat wil zeggen op het niveau van de logische structuur van data en operaties, is alles in orde.

Een tweede belangrijke wortel voor ons huidige *computational model of mind* stamt uit de formele logica van Frege, Whitehead en Russell. Het gaat hier om de vraag: hoe kunnen zuiver syntactische procedures die zich niets aantrekken van de betekenis van de symbolen waarop ze aangrijpen, toch semantisch coherente resultaten produceren? Wanneer je een Turingmachine voedt met ware expressies, hoe kun je er dan voor zorgen dat er ook weer ware expressies uit komen? In de formele logica wordt dit probleem opgelost door de syntactische structuur van formules of expressies op systematische manier te laten samenhangen met de semantische interpretatie van die expressies.

Hoe dat werkt kan met een eenvoudig voorbeeld worden toegelicht. We weten allemaal dat uit de bewering „Jan fietst en Piet loopt" kan worden afgeleid: „Piet loopt". Als de eerste expressie waar is, is de tweede het ook. Om dit resultaat te garanderen wordt nu afgesproken dat elke expressie die de syntactische structuur Z1 en Z2 heeft semantisch zowel Z1 als Z2 impliceert. Omgekeerd is voor de waarheid van de lange expressie vereist dat elk van de constituerende delen waar is. Dit is een voorbeeld van Freges compositionaliteitsprincipe: De semantische interpretatie (waarheid, etc.) van een complexe expressie kan worden afgeleid uit de semantische interpretaties

(waarheid, etc.) van haar constituenten plus de syntactische relaties die er tussen die constituenten bestaan. Om semantische coherentie te garanderen moet dus de formele taal waarin je je expressies schrijft zo'n systematische samenhang vertonen tussen syntaxis en semantische interpretatie. Dat geldt voor alle computertalen, maar niet geheel toevallig geldt het tot op zeer grote hoogte ook voor natuurlijke talen. Het zou ons zeer verbazen wanneer er een taal was waarin het systematisch het geval is dat uit „Jan fietst en Piet loopt" volgt dat „Piet fietst". Uitzonderingen bevestigen hier de regel. Uit „Jan zit in zak en as" kan niet worden afgeleid „Jan zit in as". Idioom is juist idioom, omdat het niet voldoet aan het compositionaliteitsprincipe.

Het computationele model van de menselijke geest dat zich sinds een halve eeuw ontwikkeld heeft is gebaseerd op dit principe. Ook de *language of thought*, de taal (of talen) waarin onze mentale operaties zich afspelen, vertoont deze systematische relatie tussen syntactische constituentenstructuur en semantische interpretatie. Op deze wijze kunnen puur mechanische doch structuur-afhankelijke operaties semantisch coherente produkten afleveren.

Een sinds Gödel en Turing mogelijke, maar pas sinds de jaren vijftig gebruikte vorm van semantische interpretatie, is het verwijzen naar opgeslagen algorithmen of programma's als data. De operatie wordt dan niet uitgevoerd, maar geciteerd. De geciteerde expressie kan eventueel worden veranderd, zodat een nieuwe operatie ontstaat. Op deze manier kan het programma zichzelf wijzigen, of een nieuw programma creëren. Deze mogelijkheid tot zelforganisatie binnen de klassieke architectuur staat ook centraal in SOAR, een leertheorie die ontwikkeld is door Laird, Rosenbloom en Newell (1986).

Langzamerhand is de overtuiging gegroeid dat zulke semantisch geïnterpreteerde fysische symboolsystemen noodzakelijk en voldoende zijn voor de modellering van elk intelligent gedrag. Het werk van Newell, Simon en hun medewerkers is op deze overtuiging gebaseerd. Dat werk wordt, geheel ten onrechte, nog steeds gezien als het modelleren van het langzame, bewuste, seriële menselijke nadenken, zoals dat gebeurt bij schaken of bij het bewijzen van stellingen.

Dat brengt ons bij een derde wortel van het computationele model van de menselijke geest, de ontwikkelingen in de taalkunde. In de jaren vijftig wees Chomsky op de onbeperkte *produktiviteit* van natuurlijke talen. Voor elke welgevormde zin is er een langere die ook welgevormd is. Het is de taak van de taalkunde die produktiviteit door middel van een eindig recursief mechanisme te beschrijven. Chomsky kon toen bewijzen dat dit met een recursief associatief mechanisme zoals een Markov proces, of algemener met een eindige automaat, onmogelijk was. (Zie Levelt (1973) voor definities van deze begrippen, en voor Chomsky's bewijs. Figuur 1 geeft een voorbeeld van zo'n eindige automaat)

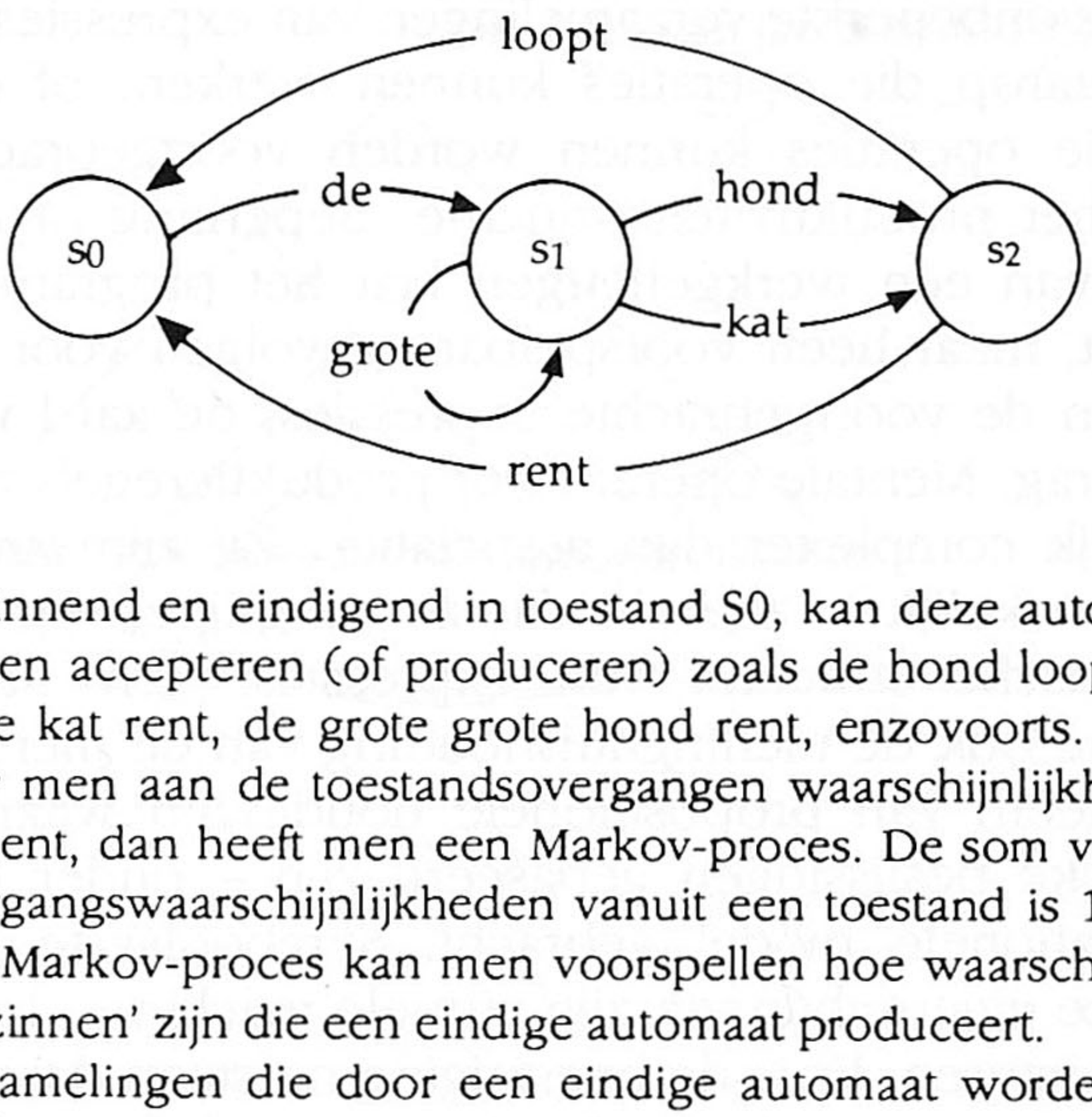
Waar het hier om gaat is dat je de zinnen van een natuurlijke taal niet kunt bouwen door elk volgende woord te kiezen alleen en uitsluitend op grond van het laatst geproduceerde woord, dus niet door het uitvoeren van een serie lokale associaties. De syntaxis van natuurlijke talen vertoont recursieve, hiërarchische eigenschappen die alleen in een wezenlijk complexere automaat gerealiseerd kunnen worden. Het gaat hier dus om een empirische eigenschap van natuurlijke talen, een bepaald type recursiviteit dat wij mensen blijkbaar in ons hebben zonder ons daarvan rekenschap te geven. Geheel onbewust binden

wij variabelen in afhankelijkheid van de recursieve, hiërarchische structuur van linguïstische expressies. Een voorbeeld daarvan is te zien in Figuur 2, waar het woord zich steeds door een ander woord in de zin gebonden wordt, afhankelijk van de hierarchische structuur van de zin.

Met deze ontdekkingen werd het behavioristische model van het menselijk taalgedrag, dat immers expliciet gebaseerd was op een associatief Markov-achtig mechanisme, terecht als volledig ontoereikend terzijde geschoven. Sindsdien is er een ware vloed van onderzoek verricht naar die abstracte linguïstische regels, die ieder van ons voortdurend en met verbluffend weinig fouten toepast. Nog dagelijks worden nieuwe regels en principes ontdekt. Er is geen sprake van dat die regels bewust worden uitgevonden, geleerd of gebruikt. De processen van spreken en verstaan zijn bovendien veel te snel voor bewuste toepassingen van regels. Een van de meest soort-eigen vermogens van de mens, de taal, blijkt een produktief recursief systeem te zijn dat niet gebaseerd is op associatieve maar op hiërarchische syntactische operaties, en dat zich grotendeels onbewust en in parallelle verwerking afspeelt. Soortgelijke vermogens zijn zelfs bij chimpansees, onze naaste buren in het dierenrijk, nooit aangetoond.

Een vierde belangrijke ontwikkeling kan ik hier slechts aanduiden. Dat is de theorie van propositionele houdingen (*propositional attitudes*). Mensen handelen op grond van wat zij weten, denken, menen, hopen, wensen, van plan zijn, enzovoorts. Dit zijn houdingen die het subject aanneemt ten aanzien van proposities, en die als oorzakelijk voor het gedrag kunnen worden opgevat. Sinds een aantal jaren is er grote vooruitgang geboekt op het punt van de formalisering, en dus het mechanisch doen bereke-

Een eindige automaat kan in een eindig aantal toestanden (Si) verkeren, waaronder een begin- en een eindtoestand, en heeft een eindig vocabulaire V van in- of uitvoersymbolen. De automaat kan van toestand wisselen wanneer een daartoegeëigend symbool wordt ingevoerd (of geproduceerd). Overgangsregels geven aan welke nieuwe toestand er wordt bereikt wanneer in een bepaalde toestand een bepaald symbool wordt ingevoerd. Een overgangsdiagram maakt dat zichtbaar.
Als voorbeeld staat hier een overgangsdiagram voor een eindige automaat met drie toestanden, S0 (begin- en eindtoestand), S1, en S2. Het vocabulaire bestaat uit zes woorden: de, grote, hond, kat, loopt, rent. Die woorden staan geschreven bij de toestandsovergangen die ze mogelijk maken.

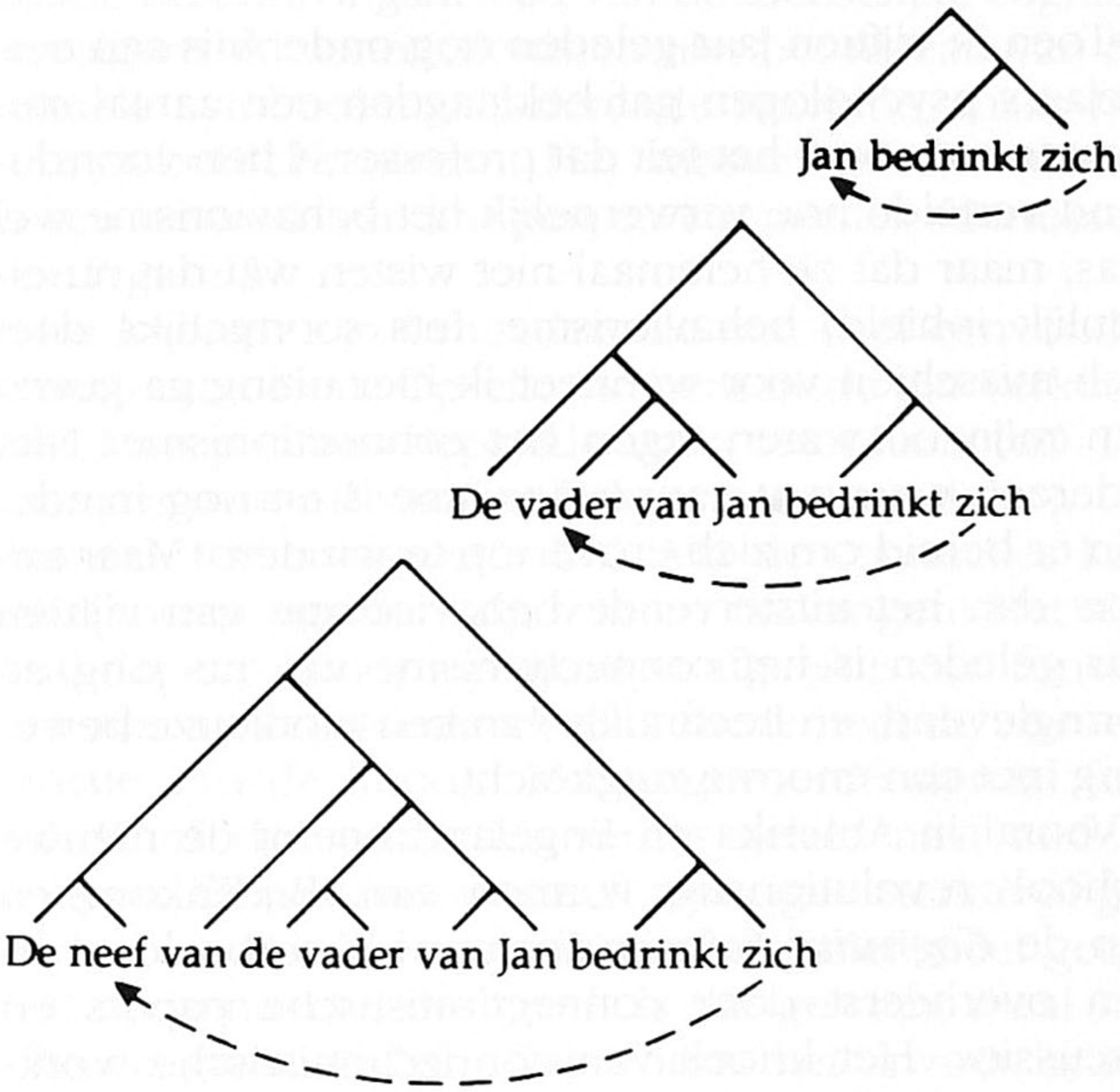


Beginnend en eindigend in toestand S0, kan deze automaat zinnen accepteren (of produceren) zoals de hond loopt, de grote kat rent, de grote grote hond rent, enzovoorts. Wanneer men aan de toestandsovergangen waarschijnlijkheden toekent, dan heeft men een Markov-proces. De som van de overgangswaarschijnlijkheden vanuit een toestand is 1. Met een Markov-proces kan men voorspellen hoe waarschijnlijk de 'zinnen' zijn die een eindige automaat produceert.
Verzamelingen die door een eindige automaat worden geproduceerd of geaccepteerd heten reguliere verzamelingen.

Figuur 1: Eindige automaten

nen van propositionele houdingen. (Levelt, 1988) Dat geeft de formele basis voor een theorie van rationeel gedrag.

Ik heb nu enkele wezenlijke eigenschappen genoemd van het computationele model van de menselijke geest, dat zich de laatste halve eeuw, hand in hand met de informatisering, heeft ontwikkeld. Kort samenvattend gaat het om een mechanisch of fysisch model van een zeer bepaalde architectuur. Mentale operaties zijn van zuiver syntactische aard, dat wil zeggen: zij zijn bepaald door de vorm, niet door de inhoud van expressies. Semantische coherentie wordt gegarandeerd door Freges principe van compositionaliteit. Er wordt een strikte scheiding gemaakt tussen eindige stelsels van mentale operaties en in principe onbeperkte verzamelingen van expressies of data waarop die operaties kunnen werken, of die door die operaties kunnen worden voortgebracht. Dat is het produktiviteitsprincipe. Beperking of expansie van een werkgeheugen laat het programma onverlet, maar heeft voorspelbare gevolgen voor de aard van de voortgebrachte expressies, de aard van het gedrag. Mentale operaties of produktieregels zijn wezenlijk complexer dan associaties. Zij zijn structuur-afhankelijk in die zin dat ze aangrijpen op de hiërarchische structuur van expressies. Ten slotte wordt nu ook de meninghuishouding van de mens – het systeem van propositionele houdingen waarop menselijke beslissingen gebaseerd zijn – onder het computationele model gebracht. Symboolsystemen met deze eigenschappen zijn virtuele machines, logische structuren. Er is de overtuiging ontstaan dat die logische structuren een verklaringsniveau *sui generis* vormen, ongeacht de implementatie van die virtuele machines. Daarmee is niet gezegd dat die implementatie irrelevant is, maar alleen dat dat niet het geëi-



Figuur 2: Recursieve syntaxis en hiërarchische structuur

gende verklaringsniveau is voor mentale processen. Net zomin als het atomaire niveau het geëigende verklaringsniveau is voor geologische processen, of het neurologische niveau voor economische processen.

Bezwaren tegen het connectionistische model

Toen ik vijftien jaar geleden nog onderwijs aan eerstejaars psychologen gaf beklaagden een aantal studenten zich over het feit dat professor X hen voortdurend vertelde hoe verwerpelijk het behaviorisme wel was, maar dat ze helemaal niet wisten wat dat nu eigenlijk inhield, behaviorisme. Iets soortgelijks doet zich misschien voor wanneer ik hier uiting ga geven aan mijn bezwaren tegen het connectionisme. Niet iedereen weet wat connectionisme is en nog minder zijn er bereid om zich erover op te winden. Maar anders dan het uitstervende behaviorisme van vijftien jaar geleden is het connectionisme van nu jong en springlevend, en heeft alles van een modieuze beweging met een enorme zuigkracht.

Vooral in Amerika en Engeland neemt de nieuwe 'school' revolutionaire vormen aan. Bijeenkomsten van de *Cognitive Science Society*, bijvoorbeeld, worden overheerst door connectionistische papers en discussies. Het krioelt van connectionistische workshops en conferenties. In veel psychologische faculteiten doen studenten weinig anders meer dan connectionistische modellen bouwen, en de tijdschriften staan bol van wat connectionistische netwerken nu weer allemaal geleerd hebben.

In Nederland wordt de soep gelukkig nog niet zo heet gegeten. Het connectionistische werk dat hier plaats vindt is van goede kwaliteit en de retoriek blijft binnen redelijke perken. Een overzichtelijke en evenwichtige inleiding in het connectionisme is te vinden

in Phaf en Murre (1989). De ontwikkelingen in het buitenland nopen echter tot waakzaamheid; vandaar dit artikel.

Het connectionistische beeld van de menselijke cognitie verschilt in vrijwel elk wezenlijk opzicht van het juist geschetste computationele model. Het geëigende beschrijvingsniveau van de menselijke cognitie is volgens de connectionisten niet het niveau van de virtuele symbool-manipulerende machine, maar het subsymbolische niveau. Het subsymbolische niveau is een netwerk van knopen en connecties daartussen (zie Figuur 3).

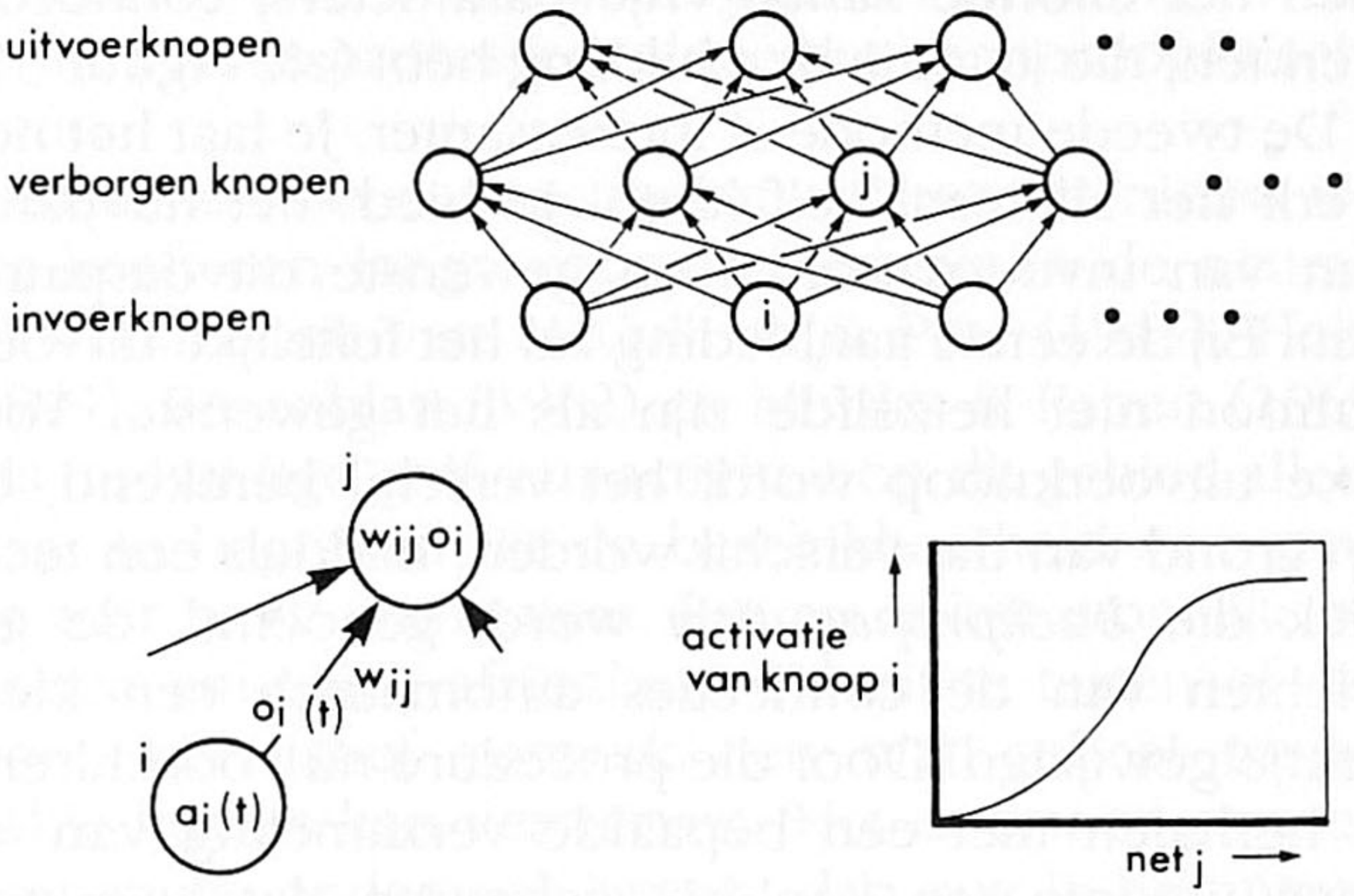
De knopen zijn eenheden die een heel eenvoudig proces uitvoeren. Op elk moment in de tijd verkeert elke knoop in een bepaalde activatietoestand (at). De activatietoestand van het hele systeem op tijdstip t is dus een toestandsvector (at). Via de connecties kan elke knoop signalen afgeven en ontvangen. Het signaal dat een knoop op tijdstip t afgeeft is een functie van zijn activatietoestand; die functie heet de uitvoerfunctie van de knoop. Het uitgevoerde signaal (ot) wordt via de connecties verder geleid naar andere knopen. Elke connectie heeft zijn eigen sterkte of gewicht (w). Dat is een positief of negatief reëel getal. Wat de connectie doorgeeft is dat gewicht maal het uitvoersignaal (w*o). Dat vormt het invoersignaal voor de ontvangende knoop. Een knoop kan op een bepaald tijdstip t signalen ontvangen via verschillende connecties. Die input signalen worden opgeteld volgens een bepaalde invoerfunctie. Meestal wordt hiervoor een quasilineaire functie gekozen: alle signalen worden lineair opgeteld, en de verkregen som wordt niet-lineair getransformeerd. Dit is om te voorkomen dat de activatie van een knoop willekeurig groot kan worden.

Het gedrag van zo'n connectionistisch netwerk, dat

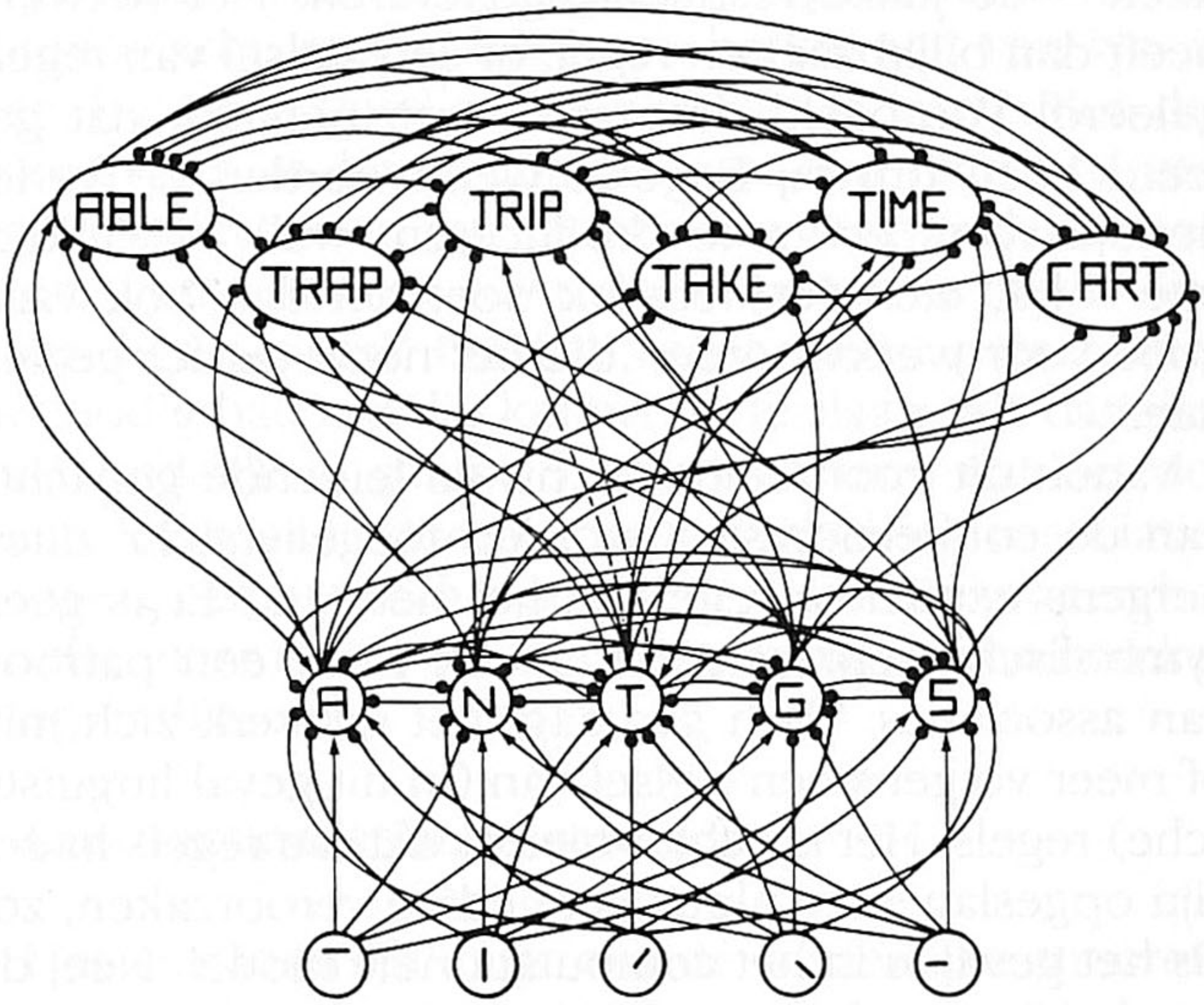
wil zeggen de verandering van de toestandsvector over de tijd, kan in principe beschreven worden met een stelsel differentiaalvergelijkingen. Meestal werkt men echter met een discrete tijdsschaal, hetgeen wat makkelijker is voor computersimulaties.

Een bepaalde deelverzameling van de knopen wordt nu gekozen als invoerknopen (*input nodes*), een andere deelverzameling als uitvoerknopen (*output nodes*). Alle andere knopen heten verborgen knopen (*hidden nodes*). Het netwerk wordt nu gestimuleerd door de invoerknopen volgens een bepaalde statistische verdeling te activeren. Dit constante statistische activatiepatroon wordt zolang gehandhaafd tot de uitvoerknopen eveneens een constante distributie van activatie gaan vertonen. Dat is de responsie van het systeem. Het netwerk zet dus een statistische invoervector i om in een statistische uitvoervector o. De relatie die zich tussen invoer- en uitvoervector vestigt wordt grotendeels bepaald door de gewichten van de connecties in het netwerk.

Er zijn globaal twee manieren om met zo'n netwerk cognitieve processen te modelleren. De oudste, maar minst interessante, is door zelf alle connecties te programmeren, dat wil zeggen door met de hand gewichten toe te kennen aan de connecties. Op die manier is het bijvoorbeeld aardig gelukt om het systeem vier-letterwoorden (zoals ABLE, TRAP, TIME) te laten herkennen. Er zijn dan invoerknopen voor ieder schuin of recht streepje in elk van de vier letters (de zogenaamde letterkenmerken). De uitvoerknopen zijn de mogelijke woorden. De kunst is nu om de connecties in het netwerk zodanige gewichten toe te kennen dat bij een bepaald invoerpatroon van letterkenmerken de juiste uitvoerknoop wordt geactiveerd. Zoiets kan gemaakt worden (McClelland & Rumelhart, 1981), maar erg opwindend is dat niet, ge-



(a) Enkele basisbegrippen



(b) Een netwerk dat 4-letter woorden herkent

Figuur 3: Connectionistische netwerken

zien het enorme aantal vrije parameters, connectiesterkten, die je tot je beschikking hebt (zie Figuur 3).

De tweede methode is interessanter. Je laat het netwerk het allemaal zelf leren. Je biedt het nu paren aan van invoerpatronen en gewenste uitvoerpatronen. Bij de eerste aanbieding zal het feitelijke uitvoerpatroon niet hetzelfde zijn als het gewenste. Voor elke uitvoerknoop wordt het verschil berekend, en op grond van dat verschil worden, middels een techniek die *backpropagation* wordt genoemd, de gewichten van de connecties automatisch een klein beetje gewijzigd. Door die procedure nu voortdurend te herhalen met een bepaalde verzameling van invoer/uitvoerparen kan het gebeuren dat zo'n netwerk de juiste associatie leert. Het kan dan soms ook bij nieuwe stimuli – die het nog niet eerder 'gezien' heeft – de juiste responsie genereren. Het netwerk heeft dan blijkbaar een regel, of een stelsel van regels geleerd. Voorbeeld daarvan is een netwerk dat geleerd heeft om bij Engelse werkwoorden de verledentijdsvorm ervan te produceren: walk – walked, bite – bit, etc. Het netwerk doet het dan ook vaak goed voor werkwoorden die het nooit eerder gezien heeft.

Vanuit dit voorbeeld kan nu de leidende gedachte van de connectionisten worden toegelicht. Er zitten nergens expliciete regels in het netwerk. Er is geen symbolische representatie, alleen maar een patroon van associaties. Toch gedraagt het netwerk zich min of meer volgens een stelsel van (in dit geval linguïstische) regels. Het is echter niet zo dat die regels intern zijn opgeslagen en aldus het gedrag veroorzaken, zoals het geval is in het computationele model. Nee, de regels zijn *emergent properties*, epifenomenen. De werkelijke gedragsdeterminanten zijn te vinden op subsymbolisch niveau; het zijn de connectiepatronen

in het netwerk. Dit is het enig juiste verklaringsniveau voor mentale processen, aldus het connectionistische credo.

Ofschoon het connectionisme allerminst nieuw is – het heeft een lange voorgeschiedenis in de neurale netwerktheorie van McCulloch & Pitts (1943), Hebb (1949), Rosenblatt (1962) en Minsky & Papert (1969) – is de huidige golf van activiteit op dit gebied alleen maar te begrijpen uit de beschikbaarheid van grote tot zeer grote computers. Pas nu is het mogelijk om binnen enigszins afzienbare tijd uit te testen of een groot associatief netwerk een niet geheel triviaal stukje kennis kan verwerven. Het doorsnee connectionistische onderzoek bestaat dan ook in het uitproberen van allerlei mogelijke kennisdomeinen: het leren van visuele patronen, het leren van logische regels, het leren van syntactische regels, het leren van fonologische regels, het leren herkennen van woorden, het leren optellen, en wat dies meer zij. Men definieert het te leren domein, zet een leerplan, een presentatierooster op, stopt alles 's avonds in een grote computer, en kijkt de volgende ochtend hoe goed het netwerk het kan, en hoeveel presentaties het nodig had om die kennis op te slaan. Als dat allemaal redelijk is gelukt concludeert men dat het domein leerbaar is door een subsymbolisch associatief netwerk, en dat er dus ook voor dat domein geen behoefte meer is aan een klassieke symbolische berekeningsarchitectuur.

Connectionistische retorica

Het is natuurlijk zaak om de wezenlijke theoretische kwesties goed te scheiden van de retoriek, maar het kan geen kwaad om bij die retoriek even stil te staan. Het is namelijk niet altijd even eenvoudig om

feit en fantasie uit elkaar te houden in de geschriften van connectionisten. De vlag dekt vaak de lading niet. De verpakking van het connectionisme bestaat niet zelden uit een stroom van megalomane beweringen over het eigen kunnen, en kleineringen betreffende de klassieke architectuur. Beide zijn veelal ongefundeerd dan wel aantoonbaar onjuist. Connectionisme zou zijn een „theory from which the multiplicity of conceptual theories can be seen to emerge" (Hunter, 1988); „It is likely that connectionist models will offer the most significant progress of the past several millenia on the mind/body problem." (Smolensky 1988) Connectionistische netwerken zouden „effectively Turing machines" zijn (Elman, 1989), een bewering die niet uit de premissen volgt (zie hier onder). Het opheffen van de scheiding tussen programma en data wordt voorgesteld als een theoretische doorbraak (dat is het ook, maar in negatieve zin). Er wordt met enthousiasme gewezen op het chaotische karakter van connectionistische systemen; het zou ze zo geschikt maken voor zelf-organisatie. Connectionistische modellen zouden nieuwe hoop bieden voor het begrijpen van verschijnselen als intuïtie, *common sense*, creativiteit, bewustzijn, zelfbewustzijn. Kortom, veel geloof en hoop. En ook veel haat. De klassieke architectuur en de fenomenen die zij beoogt te verklaren worden schamper terzijde geschoven: „recursive processing (is not) of the essence of human computation" (Rumelhart & McClelland, 1986, p.119); „there is no induction problem. The child need not figure out what the rules are, not even that there are rules." (Rumelhart & McClelland, 1986, p.267) Er wordt bij herhaling en ten onrechte beweerd dat symbool-verwerkende modellen slechts symbolen en regels toelaten die bewust toegankelijk zijn, dat ze slechts sequentiële, geen parallelle ver-

werking toelaten, dat ze niet in staat zijn afwijkend gedrag, onregelmatigheden, fouten te voorspellen of te verklaren, enzovoorts.

Wetenschapsbeoefening zonder retoriek bestaat niet. Maar het wordt wel zorgelijk wanneer we, zoals thans op grote schaal gebeurt, onze studenten volstoppen met illusies, wanneer we ze opvoeden als analfabeten die geen weet hebben van wat er op het gebied van computationele theorie de laatste halve eeuw aan fundamentele inzichten is verworven, en wanneer we ze het idee geven dat empirisch onderzoek bestaat in het testen of het model iets leren kan in plaats van of mensen iets leren kunnen. Er bestaat geen noemenswaardige empirische (experimentele) connectionistische psychologie.

Laten we de retoriek echter voor wat zij is, om enkele theoretische en empirische problemen te analyseren die zich voordoen met een connectionistisch model van de menselijke geest.

Enkele theoretische en empirische problemen

Het connectionisme zet elke essentiële verworvenheid van meer dan een halve eeuw computationele theorie overboord. De eerste is dat mentale processen beschouwd kunnen worden als syntactische operaties op symbolische representaties. De tweede is het inzicht dat de semantische coherentie van een syntactische machine gebaseerd moet zijn op Freges compositionaliteitsprincipe, en dus op de hiërarchische constituentenstructuur van symbolische expressies. De derde is het onderscheid tussen een eindig opgeslagen programma en een in beginsel onbeperkte verzameling data (het *stored program concept*), en daarmee nauw samenhangend tussen een beperkte computationele structuur en een schier onbeperkt

kennisbestand. En daarmee verdwijnt ook een vierde verworvenheid: het inzicht dat de menselijke cognitie met beperkte computationele middelen onbeperkte kennisbestanden kan voortbrengen of interpreteren (de produktiviteitsassumptie). Ten slotte wordt de veronderstelling verworpen dat er echte verklaringen mogelijk zijn op symbolisch niveau, met name in termen van propositionele houdingen zoals wensen, meningen, doelstellingen, en overtuigingen. Dit zouden slechts *emerging properties* zijn, epifenomenen. De scheiding der geesten is hier nagenoeg volkomen. Toch wil ik een aantal problemen aanduiden die zich onherroepelijk voordoen bij het nemen van deze drastische stappen.

Een eerste probleem betreft de leerbaarheid van kennisbestanden. We hebben gezien dat connectionisten hevig geïnteresseerd zijn in de leerbaarheid van allerlei kennisdomeinen. Kan men een netwerk een bepaalde verzameling woorden, visuele objecten, zinnen, logische bewerkingen leren? Die leerbaarheidsvraag wordt steeds beantwoord door het maar uit te proberen op een computer. Dit is een buitengewoon onprincipiële weg. (Levelt, 1990a) Men mag namelijk uit de leerbaarheid van een bepaald kennisbestand nooit concluderen dat een groter kennisbestand van dezelfde soort ook leerbaar is. Zeg dat een connectionistisch model een taal kan leren met zinnen van het type „Als Jan zegt dat het regent jokt hij". Er is dan geen enkele garantie dat het model ook zinnen aankan zoals „Als Jan zegt dat Piet zegt dat het regent jokt hij". Maar misschien kan een veel groter netwerk dat ook nog. Helaas is er dan weer geen enkele garantie dat dat netwerk zinnen aankan van het type „Als Jan zegt dat Piet zegt dat Kees zegt dat het regent jokt hij", enzovoorts. Kortom, we kunnen op deze manier nooit te weten

komen of het netwerk een taal kan leren die onbeperkte recursie van deze aard toelaat (zo'n taal heet een contextvrije taal).

Computersimulatie van een model is geïnstitutionaliseerde luiheid. Wat echt nodig is, is een bewijs of iets leerbaar is of niet, en als het niet leerbaar is, waar dan de asymptoot ligt van leerbaarheid (bijvoorbeeld bij twee-, drie- of viervoudige recursie). In de klassieke traditie, maar interessant genoeg ook bij de voorlopers van het connectionisme, McCulloch, Pitts, Rosenblatt, Minsky en Papert, was dat ook de normale gang van zaken. Men bewees of een verzameling leerbaar was of niet.

Er is in de klassieke traditie veel bekend geworden over de leerbaarheid van allerlei verzamelingen, zoals die van contextvrije en andere talen. Het ontbreekt echter volkomen aan leerbaarheidsbewijzen binnen het hedendaagse connectionisme. Er is dus geen basis geschapen voor generalisaties. Elke bewering dat een connectionistisch model X kan leren, waar X een niet-reguliere verzameling is, is bluf.

Nauw hiermee in verband staand is het gebrek aan inzichtelijkheid van een resultaat. Wanneer een connectionistisch model iets geleerd heeft kan men zeggen „Het heeft X geleerd", en „Het heeft er zolang over gedaan", maar men weet nog steeds niet waarom het X heeft kunnen leren of waarom het Y niet heeft kunnen leren. We ontdekken op deze manier geen verklaringsprincipes. Een connectionistisch model is *mutatis mutandis* net zo handig als een één-op-één plattegrond van een stad.

Het ontbreken van leerbaarheidsbewijzen hangt weer samen met een ander defect. Om te weten wat een mechanisme kan leren moet men eerst weten wat het kan genereren. Dit vereist enige toelichting. Het mogelijke kennisdomein van een klassieke archi-

tectuur wordt bepaald door de eindige verzameling van regels of operaties, het programma. Ik noemde eerder Chomsky's bewijs dat een eindige automaat geen natuurlijke taal kan representeren. Je kunt zo'n automaat niet programmeren op zo'n manier dat het elke zin van het Nederlands kan produceren, zonder ooit een ongrammaticale rij woorden voort te brengen. Als een kennisdomein, zoals een taal, niet gerepresenteerd kan worden, dan kan het natuurlijk al helemaal niet geleerd worden. Maar het omgekeerde geldt merkwaardig genoeg niet. Wanneer een automaat een bepaald kennisdomein in principe wel kan representeren hoeft het dat kennisdomein nog helemaal niet te kunnen leren.[2]

Welke kennisdomeinen kan een connectionistisch netwerk representeren? Op deze vraag kan, in eerste instantie, een eenvoudig antwoord worden gegeven. Een connectionistisch netwerk is een eindige automaat, want het bestaat uit een eindig aantal knopen die elk in een eindig aantal toestanden kunnen verkeren. Het netwerk als geheel kan dus slechts in een eindig aantal onderscheidbare toestanden verkeren. Als eindige automaat kan zo'n netwerk slechts reguliere verzamelingen representeren (zie de tekst in Figuur 1). Het is dan principieel uitgesloten dat complexere dan reguliere verzamelingen, zoals natuurlijke talen of predikatenlogica, in een connectionistisch netwerk kunnen worden gerepresenteerd.

Hiermee lijkt het doek gevallen te zijn, maar zo eenvoudig is het niet. In de eerste plaats hoeft de activatietoestand van een knoop geen discrete variabele te zijn (met een eindig aantal waarden). Onlangs hebben Hornik, Stinchcombe & White (1989) bewezen dat netwerken met verborgen knopen en een continue activatievariabele in staat zijn tot het simuleren van elke meetbare functie. Dit is ongetwijfeld een

belangrijk resultaat. Het is voor het eerst dat er ten aanzien van het generatieve vermogen van netwerken iets wordt bewezen in plaats van alleen maar beweerd. De invloedrijke connectionist Elman (1989) concludeerde uit dit resultaat echter meteen dat connectionistische netwerken „are effectively Turing machines". Dat wil zeggen, zij zouden tot elke symbolische berekening in staat zijn. Levelt (1990b) toonde echter aan dat deze conclusie niet volgt uit het resultaat van Hornik et al.

Maar ook als men zich beperkt tot netwerken met discrete toestandsvariablelen, zou men als connectionist het volgende kunnen antwoorden: „Het representeren van een complexere verzameling vraagt ook bij een klassieke architectuur om een onbeperkt geheugen, een oneindig lange tape in de Turing machine. Dat mensen zo'n oneindig lange tape in hun hoofd hebben kan niet anders zijn dan een theoretische idealisatie; dat moet ons ook worden gegund. Met andere woorden, we willen de mogelijkheid hebben het netwerk onbeperkt te laten uitdijen."

Die reactie is fair, maar nu ontstaan er weer nieuwe moeilijkheden voor het connectionisme. Het is uiteraard niet de bedoeling uit te gaan van een oneindig groot netwerk; daarvoor heb je namelijk een oneindig aantal vergelijkingen nodig. Waar behoefte aan is, is een procedure om het netwerk steeds net zoveel te vergroten als nodig is om een bepaalde berekening uit te voeren. Als het netwerk bijvoorbeeld net te klein is om zinnen te herkennen zoals „Als Jan zegt dat Piet zegt dat het regent jokt hij", moet er zoveel kunnen bijgroeien dat ook dat weer gaat.

Maar hier wreekt zich wat juichend als revolutionair succes is binnengehaald: het opheffen van de scheiding tussen programma en data. Wanneer een netwerk iets geleerd heeft, en je zet er een stukje aan,

dan kan het geleerde weer verloren gaan. Met andere woorden vergroting van het netwerk verandert niet alleen de grootte van het 'werkgeheugen', de omvang van de *resources*, maar ook de computationele architectuur – de geleerde regels.[3] Dat nu overkomt je niet met een klassieke architectuur. Dit brengt me bij het volgende punt:

Connectionistische netwerken zijn intolerant voor kennisvermeerdering. Wanneer een connectionistisch netwerk verzameling X geleerd heeft, en je leert het vervolgens verzameling Y, dan is verzameling X weer vergeten. Dit is een direct gevolg van het opgeven van Turings principe, het scheiden van programma en data. De enige manier waarop een netwerk bij kan leren is door al het oude weer opnieuw mee te leren – voorwaar een voortreffelijk model van het menselijk leervermogen!

Al even hopeloos is het een connectionistische verklaring te geven voor het directe leren van regels, een van onze belangrijkste vormen van kennisvermeerdering. Als de telefoonnummers in mijn stad vanaf morgen een extra decimaal krijgen, het begincijfer 2, en ik verneem die regel, dan kan ik die direct toepassen zonder alle nummer/naam associaties opnieuw te leren, resectievelijk op te zoeken in een nieuw telefoonboek. Een connectionistisch netwerk moet geheel opnieuw getraind worden. Een groot deel van onze handelingskennis bestaat echter uit zulke eenmalig geleerde regels. (Zie Levelt 1990a voor meer over de hier gesignaleerde problemen.)

Connectionistische modellen zijn indifferent voor semantische coherentie. Dit is op alle fronten het geval. Als een netwerk geleerd heeft dat de zin „Jan fietst en Piet loopt" waar is, weet het niet dat „Jan fietst" waar is, of dat „Piet loopt" waar is. Wanneer het beide inferenties geleerd heeft weet het vervolgens

weer niet dat uit „Piet fietst" en „Jan loopt" volgt dat „Piet fietst en dat Jan loopt". Dat komt doordat regels en data niet gescheiden zijn. Wanneer die abstracte regel, namelijk dat uit P&Q zowel P als Q volgen, na veel training op heel veel zinnen, ten slotte toch redelijk geleerd is, weet het nog steeds niet dat uit „Jan fietst en Piet loopt en Marietje werkt" volgt dat „Jan fietst". Wat er uit P&Q&R volgt moet weer helemaal geleerd worden. Dit is uiteraard het gevolg van het buiten beschouwing laten van de constituentenstructuur van expressies. Er is geen basis om uit gelijksoortige syntactische structuur gelijksoortige semantische inferenties te maken. Veel erger nog: voor hetzelfde geld kan men het netwerk leren om uit „Jan fietst" en „Piet loopt" en „Marietje werkt" te concluderen dat Jan niet fietst. Hetzelfde netwerk concludeert dan uit de tweeledige zin dat Jan wel fietst, en uit de drieledige zin dat Jan niet fietst.

Dit soort voorbeelden kan *ad libitum* worden uitgebreid. Ze tonen aan dat het connectionisme neutraal is waar het dat juist niet zou moeten zijn als een *model of mind*: het is neutraal ten aanzien van semantische coherentie; er is niets dat semantische anarchie verbiedt. Het zou toch een eerste taak van zo'n model moeten zijn te verklaren waarom de menselijke geest semantisch niet arbitrair functioneert? Dat is precies wat de klassieke architectuur doet dank zij het compositionaliteitsprincipe en de scheiding van programma en data. (Zie Fodor & Pylyshyn (1988) voor een grondige discussie van deze kwestie.)

Enkele andere problemen. Er is een veelheid van andere theoretische problemen, zoals de onmogelijkheid types en tokens uit elkaar te houden (Prince & Pinker, 1988), de onoverkomelijke problemen met het binden van variabelen (Zodat er geen systematische basis is om de referentie van zich in „Jan be-

drinkt zich" en in „De vader van Jan bedrinkt zich" correct te interpreteren, zie Figuur 2 – hoe men zich menselijke cognitie zonder de mogelijkheid variabelen te binden voorstelt is mij een raadsel), de onmogelijkheid om op systematische wijze de logica van verschillende propositionele houdingen te onderscheiden (Je kunt niet tegelijk twee elkaar uitsluitende dingen geloven, maar wel wensen), het ontbreken van een mechanisme voor aandachtscontrole, enzovoorts.

Ik wil hier echter nog een empirische kwestie noemen, die me zeer ter harte gaat. Zoals al opgemerkt, bestaat het 'empirisch' connectionistisch onderzoek in het uitproberen of domein X geleerd kan worden door het netwerk. Wanneer het antwoord enigszins positief is, blijft dan de centrale vraag of mensen X op dezelfde manier leren. Connectionisten stellen zich die vraag meestal niet. Maar die paar keer dat dat is nagegaan (door onderzoekers buiten het 'kamp') was het antwoord een blamage voor het connectionistische model. Het best uitgewerkte geval is dat van de verledentijdsvormen, waar het connectionistische model dat Rumelhart & McClelland (1986) hadden voorgesteld er werkelijk hopeloos naast bleek te zitten. Niet alleen leerde het van alles verkeerd, maar het leerproces liep op essentiële punten anders dan dat van kinderen. (Pinker & Prince, 1988) Eén zo'n essentieel punt is dat een kind zijn gedrag slechts zou dienen te veranderen wanneer het geconfronteerd wordt met andere (nieuwe) invoer-contingenties. Dat is aantoonbaar onjuist. Algemener gesteld: connectionistisch leren is strikt frequentieafhankelijk. Het is inmiddels uit empirisch onderzoek genoegzaam bekend dat dit niet geldt voor menselijk leren.

Wat doen we ermee?

Gegeven het feit dat een connectionistisch model van de menselijke geest de meest wezenlijke kenmerken van het cognitief functioneren niet kan behandelen, doet zich nu de vraag voor: waar kunnen connectionistische netwerken dan nog nuttig voor worden gebruikt?

Op deze vraag heb ik twee reacties, één pragmatische en één meer principiële. De pragmatische reactie is deze: laat ieder ermee doen waar hij zin in heeft. Er zijn stellig beperkte problemen die zich met een connectionistisch model laten behandelen, zoals statistische inferentie, opscherpen van vage fotografische beelden, *content-addressable storage* van vaste databestanden (waar overigens de klassieke oplossingen nog steeds veel beter zijn). De wetenschap is een soort anarchistische markt, en we merken wel welk produkt zijn waarde zal behouden. Deze reactie is echter niet helemaal bevredigend. We worden er tenslotte voor betaald om goed na te denken, vandaar een meer principiële reactie: welke theoretische plaats kan er voor connectionistische netwerken worden ingeruimd in een theorie van de menselijke cognitie? Mijns inziens kan dat geen andere zijn dan die van een potentieel implementatiemedium. (Ik sluit me hier aan bij de opvatting van Fodor & Pylyshyn, 1988.) Het is betrekkelijk, maar niet volkomen irrelevant hoe een cognitief model wordt geïmplementeerd. Zo zijn veel van de bezwaren die connectionisten uiten ten aanzien van de klassieke architectuur, met name dat die slechts geschikt zou zijn voor langzame seriële verwerking, geen tolerantie zou hebben voor ruis, etc., in feite bezwaren tegen de implementatie ervan in Von Neumann machines. De meeste moderne cognitieve modellen zelf, zoals die voor

taalverstaan en taalproduktie, hebben een volkomen parallelle architectuur (Levelt, 1989), die zich derhalve wat moeizaam laten implementeren in Von Neumann machines. Het is zeker denkbaar dat aspecten van zulke architecturen zich beter laten implementeren in een connectionistisch netwerk, en dat is de moeite van het uitproberen waard.

Ik vind het overigens wat vroeg om zelfs op dit punt grote verwachtingen te koesteren. Wat namelijk vrijwel onmogelijk is in een gedistribueerde netwerkrepresentatie is *multiple tasking*, de mogelijkheid om eenzelfde netwerk parallel meerdere taken tegelijk te laten uitvoeren. Connectionisten hebben de mond vol van 'massieve parallelle verwerking', maar wanneer een netwerk is geprogrammeerd voor de ene taak op het ene kennisbestand, dan gaat die kennis weer teloor, zoals we zagen, wanneer het een andere taak op een ander kennisbestand moet leren uitvoeren. De enige realistische oplossing is dus om elke vaardigheid, elk deelproces, in een eigen netwerkje onder te brengen. Die netwerkjes kunnen dan onafhankelijk en parallel aan de gang. Een moeilijkheid daarbij is dan weer dat de kennis opgeslagen in netwerk X niet portable is naar netwerk Y. Het ene netwerk kan de kennis in een ander netwerk niet 'lezen'; er ontstaan enorme informatietransmissie-problemen, zelfs in zo'n modulair opgebouwd systeem van netwerken.

Er zijn ook andere redenen om er aan te twijfelen of zo'n netwerk handig is als implementatie. Een gevleugeld woord van connectionisten is *brain-style modelling*. Connectionistische netwerken zouden lijken op neuronale netwerken, dus op het cerebrale substraat waarin alle cognitie in laatste instantie is geïmplementeerd. We weten inmiddels dat dat in de verste verte niet het geval is. (Zie bijvoorbeeld wat de

connectionistische voorman Smolensky (1988) daar nu over zegt. Zie ook hoe Crick (1989) die neuronale pretenties onderuit haalt – met name de neurologische realiteit van *backward propagation*, een voor het connectionisme essentieel begrip. Die notie is strijdig met het eenrichtingsverkeer in de signaaltransmissie van neuronen.) Een beetje meer *brain-style modelling* zou zeker geen kwaad kunnen. Waarom laten connectionisten als enige operatie in hun netwerken additie toe, het al of niet lineair optellen van activatiestromen? Echte neuronen zijn tot veel meer in staat, met name tot allerhand logische schakelingen. Je kunt die logische schakelingen vermoedelijk wel allemaal programmeren in een connectionistisch netwerk, maar dat is een louter door het formalisme veroorzaakte rijstebrijberg. Implementatie van een beetje complex gedrag vraagt om logische schakelingen. Maar zulke schakelingen zijn taboe voor connectionisten.

Conclusie

Het beschikbaar komen van zeer groot rekentuig heeft een oude intellectuele traditie in het westerse denken nieuw leven ingeblazen, het associationisme. Maar de huidige reïncarnatie, het connectionisme, is onderhevig aan dezelfde kritiek waaraan vroegere varianten, zoals die van Hume, alsook het behaviorisme, blootstonden. Als model van de menselijke geest is het connectionisme geen alternatief voor wat ik de klassieke computationele architectuur heb genoemd. Het beste wat we hopen kunnen is dat het nieuwe speelgoed geschikt is voor het modelleren van een aantal deelaspecten van de menselijke cognitie, met name die waar het kennisbestand beperkt, en niet van recursieve aard is. Ook laten connectionistische

netwerken zich wellicht gebruiken als min of meer handige implementatiemedia voor een aantal cognitieve operaties. Maar meer moet men er toch niet van verwachten.

Literatuur

Crick, F. (1989): The recent excitement about neural networks, *Nature*, 337, 129-132.

Elman, J.L. (1989): *Representation and Structure in Connectionist Models*. CRL Technical Report 8903.

Fodor, J.A. & Pylyshyn, Z.W. (1988): Connectionism and cognitive architecture: A critical analysis. *Cognition*, 28, 3-71.

Hebb, D.O. (1949): *The organization of behavior*. New York: Wiley.

Hornick, K., Stinchcombe, M. and White, H. (1989): *Multilayer feedforward networks are universal approximators*. Discussion Paper 88-45R. Department of Economics, UCSD.

Hunter, L.E. (1988): Some memory, but no mind, *Behavioral and Brain Sciences*, 11, 37-38.

Laird, J.E., Rosenbloom, P.S. & Newell, A. (1986): Chunking in Soars: The anatomy of a general learning mechanism. *Machine Learning*, 1, 11-46.

Levelt, W.J.M. (1973): *Formele grammatica's in linguistiek en taalpsychologie*. 3 dln. Deventer: Van Loghum Slaterus.

Levelt, W.J.M. (1988): Onder sociale wetenschappen. Toegelicht aan psychologie, economie en taalkunde. *Mededelingen KNAW*, Deel 51, no 2.

Levelt, W.J.M. (1989): *Speaking: From Intention to Articulation*. Cambridge, Mass.: MIT Press.

Levelt, W.J.M. (1990a): Are multilayer feedforward networks effectively Turing machines?, *Proceedings of Conference on Domains of Mental Functioning:*

Attempts at a Synthesis. Psychological Research, 52.

Levelt, W.J.M. (1990b): On learnability, empirical foundations, and naturalness. Commentary on S.J. Hanson and J. Burr, What connectionist models learn: Learning and representation in connectionist networks. *Behavioral and Brain Sciences*, 13.

McClelland, J.L. & Rumelhart, D.E. (1981): An interactive activation model of context effects in letter perception: Part 1. An account of basic findings. *Psychological Review*, 88, 375-407.

McCulloch, W.S. & Pitts, W. (1943): A logical calculus of the ideas immanent in neural nets. *Bulletin of Mathematical Biophysics*, 5, 115-137.

Minsky, M.L. & Papert, S.A. (1969): *Perceptrons*. Cambridge, Mass.: MIT Press.

Phaff, H. en Murre, J. (1989): Cognitie onder de microscoop. In: C. Brown, P. Hagoort en Th. Meijering (Red.): *Vensters op de geest. Cognitie op het snijvlak van filosofie en psychologie*. Utrecht: Grafiet.

Pinker, S. & Prince, A. (1988): On language and connectionism: Analysis of a parallel distributed processing model of language acquisition. *Cognition*, 28, 73-193.

Prince, A. & Pinker, A. (1988): Subsymbols aren't much good outside of a symbol-processing architecture. *Behavioral and Brain Sciences*, 11, 46-47.

Rosenblatt, D.E. (1962): *Principles of neurodynamics*. New York: Spartan.

Rumelhart, D.E. & McClelland, J.L. (Eds.) (1986): *Parallel distributed processing*. Vol. I. Cambridge, Mass.: MIT Press.

Smolensky, P. (1988): On the proper treatment of connectionism. *Behavioral and Brain Sciences*, 11, 1-42.

Turing, A.M. (1936): On computable numbers, with

an application to the Entscheidungsproblem. *Proceedings of the London Mathematical Society*, 42, 230-265.

Noten

1. Deze bijdrage verscheen in enigszins gewijzigde vorm in C. Brown, P. Hagoort & Th. Meijering (red.) (1989): Vensters op de geest. Utrecht: Grafiet.

2. Op dit punt maken connectionisten het zich moeilijker dan nodig is. Hun eerste vraag is of een kennisdomein in een netwerk gerepresenteerd kan worden. Maar als iets niet leerbaar is, kan het nog best representeerbaar zijn.

3. In een recent, ongepubliceerd artikel stelt Ash (1989) een methode voor om een netwerk voorzichtig te laten groeien zonder het geleerde weer kwijt te raken. In een beperkt aantal computersimulaties blijkt dat ook te lukken. Er is echter niets bekend over de generaliseerbaarheid van dit resultaat.