

## COMPUTERSIMULATIE EN HOOFDREKENEN\*

Marc DE MEY, Eerstaanwezend Assistent

Seminarie logica en kennisleer RUG

Simulaties vormen slechts een relatief klein onderdeel van de grote verzameling onderzoeken waarbij in de psychologie computers worden betrokken. In overeenstemming met een indeling van Newell en Simon (1963, pp. 364-366) onderscheiden we vier toepassingsgebieden :

- gebruik van de computer als "rekenmachine";
- gebruik van computers voor de controle van psychologische experimenten (genereren van "stimuli", registreren en verwerken van "responses");
- gebruik van de computer voor simulatie van psychologische processen;
- computersystemen als modellen voor menselijke informatieverwerking.

Statistische methoden zijn reeds lang in gebruik onder psychologen. Het gebruik van computers laat nu evenwel onderzoeken toe die voorheen alleen reeds door de omvangrijkheid van de berekeningen onuitvoerbaar bleken. Dit gebruik van de computer als "rekenmachine", evenals het veelvuldig gebruik van computers voor de controle van psychologische experimenten, laten we hier buiten beschouwing. De hierna vermelde onderzoeken werden ontwikkeld als simulaties van cognitieve processen en worden verder uitgewerkt en geïnterpreteerd binnen een theorie waarvoor bepaalde computersystemen als model fungeren.

### *Computersimulatie.*

Simulatie is bepaald geworden als "the act of representing some aspects of the real world by numbers or symbols which may be easily manipulated to facilitate their study" (McCoy) en ook als "the development and use of models for the study of dynamics of existing or hypothesized systems" (McLeod, 1968, p. 5). Het valt op dat in deze definities geen duidelijk onderscheid wordt uitgedrukt ten overstaan van wiskundige modellen. Wellicht is er geen strikt onderscheid te maken tussen wiskundige modellen en simulatie-programma's. Simulatie vormt het tussenstadium in de ontwikkeling van een vage verbale theorie naar een exact wiskundig model. Programma's die bepaalde gedragingen "simuleren" of "nabootsen" worden door psychologen beschouwd als theorieën betreffende die gedragingen. Wat de formulering van theorieën in termen van programma's voor heeft op puur verbaal gestelde theorieën is : grotere graad van specificatie, zowel van de gepostuleerde processen als van de producten die uit dergelijke processen kunnen voortsprui-

ten. Dit wordt onder meer door Frijda benadrukt : "La simulation des processus psychologiques au moyen des ordinateurs, on l'a souvent répété, sert à formuler des théories d'une façon explicite et précise; elle sert à prouver que ces théories suffisent à expliquer les phénomènes; elle sert à produire les conséquences de ces théories, qui peuvent alors être comparées aux données empiriques" (Frijda, 1968, p. 109). De stijgende populariteit van simulatie in de psychologie is een onderdeel van een algemene tendens die door Oettinger wordt opgemerkt, namelijk : dat "in disciplines where mathematics is not the prevailing mode of expression the language of computer programs serves increasingly as the language of science" (Oettinger, 1969, p. 201). Noteer dat het hierbij steeds "programma's" betreft. Laughery en Gregg (1962) en ook Neisser (1967) hebben erop gewezen dat we bij simulatie de computerprogramma's niet mogen verwarren met de computer waarop de programma's worden gerealiseerd : "Note that the computer program (not the computer itself) fully embodies the 'theory'" (Laughery & Gregg, 1962, p. 266). Het programma wordt geacht overeen te stemmen met een theorie omtrent een of ander psychologisch proces en de simulatie is a.h.w. een mechanisch "inferentie-systeem" voor het bepalen van de voorspellingen inherent aan de betrokken theorie. Computersimulatie in de psychologie houdt dus geenszins verband met de vooropstelling dat het menselijk brein verwantschap vertoont met computers. Het simuleren als modelleren van een psychologisch proces enerzijds en het modelleren van de "mind" in termen van computer-concepten anderzijds, zijn duidelijk te onderscheiden en kunnen elk afzonderlijk en onderling volstrekt onafhankelijk worden nagestreefd.

#### *Getallenreeksen en geheugen.*

Ongetwijfeld zijn wij allen vertrouwd met het fenomeen "vergeten". Veelal heeft dit begrip een negatieve connotatie. Vergeten komt neer op "kwijt geraken", verlies van kennis. We vergeten de titel van een belangrijk boek, we vergeten de structuur van een bewijsvoering, we vergeten een afspraak, een adres, een datum, een telefoonnummer, ... kortom, we vergeten vele nuttige dingen. Een vraag is of we dit wel zo negatief moeten zien. Sommigen hebben immers gepoogd vergeten te zien als een soort opruimingsproces ("garbage collecting" in het computer-jargon) via hetwelke ons informatie-verwerkend-systeem zich ontdoet van relatief weinig gebruikte kenniselementen. Er is ook op gewezen dat bij het programmeren van computers een soort "trade off" moet worden in acht genomen tussen complexiteit van de data-structuur (geheugen) en programma's (denken) (Feigenbaum, 1970, p. 459). In termen van psychologische entiteiten zou dit neerkomen op : hoe groter de complexiteit van het geheugen hoe eenvoudiger de denkpatronen, en ook omgekeerd : hoe eenvoudiger het geheugen hoe groter de complexiteit van de denkpatronen. Houdt de capaciteit van ons geheugen verband met de wijze waarop we denken ? Hoe is de onderlinge afhankelijkheid tussen denken en geheugen ? Uit onze gegevens blijkt dat de wijze waarop we bepaalde dingen vergeten bepaalde karakteristieken weerspiegelt van de wijze waarop we denken. Eén vorm van vergeten verdient daarbij speciale aandacht : het vergeten dat we autonoom ontdekken. Soms maakt onze omgeving

ons erop attent dat we een of ander vergeten : het wordt ons gezegd of het blijkt uit omstandigheden. Bij het thuiskomen vinden we op onze schrijftafel de brief die we hadden moeten meenemen. Laten we die vormen voorlopig buiten beschouwing en beperken we ons tot het "gedeeltelijk vergeten". We weten nog dat we iets moeten doen maar we zijn vergeten wat het juist is. We weten nog dat we een afspraak hadden maar we weten niet meer waar of met wie. We beëindigen een parentesis in onze uiteenzetting en ... we weten niet langer waar we gebleven waren : we zijn zelf de draad kwijt. Ook dat is wellicht voor velen een vertrouwde ervaring. Onze pogingen tot simulatie van bepaalde mentale processen betrokken bij het oplossen van relatief eenvoudige rekenkundige problemen wijzen uit dat een dergelijk vergeten helemaal niet accidenteel is. Het raakt aan een essentiële karakteristiek van ons gedrag : de hiërarchische organisatie van onze handelingen, intern handelen zowel als extern, denken evenzeer als motoriek.

Vrij sterke versies van de Turing-test voor computers worden gevormd door bepaalde opgaven van traditionele intelligentietests. Computerprogramma's die ongeveer als mensen op dergelijke testopgaven reageren maken een kans als beschrijvingen van de menselijke intelligentie. Geïnspireerd door een studie van Simon en Kotovski (1963) over het voortzetten van letterreeksen hebben wij aanvankelijk het voortzetten van getallenreeksen tot specifiek simulatiedomein gekozen. Simon en Kotovski kozen als taken de items van de *Thurstone Letter Series Completion Test* waarbij de proefpersoon een reeks letters wordt aangeboden waarvoor hij een volgende dient te vinden, voorbeelden :

cadaeafa-  
rscdstdetuef-

Getallenreeksen komen voor in diverse intelligentietests, o.a *Meili*, *Coetsier* en *Amthauer IST*. Zij zijn uiteenlopend van aard. We beperken ons hier tot een type dat steunt op het principe van de rekenkundige reeks. Een paar voorbeelden :

3	7	11	15	19	23	..	..
17	16	19	15	21	14	..	..
1	3	6	12	24	44	75	.. ..

De opgave luidt dergelijke reeksen met een of twee getallen verder te zetten.

Op basis van analyse's van door proefpersonen gerapporteerde methoden ontwikkelden we twee programma's die psychologisch relevant blijken.

Het eerste programma bevat gewoon de lijst van gerapporteerde oplossingsmethoden voor de reeksen van *Meili*. Een voorbeeld van zo'n oplossingsmethode is voor

17	16	19	15	21	14
----	----	----	----	----	----

een ontbinding van de reeks in twee elementaire rekenkundige reeksen :

17	19	21
16	15	14

Stellen we de reeksen algemeen voor als reeksen van  $n$  getallen  $a(0) \dots a(n-1)$ , dan blijkt in dit geval het verschil tussen  $a(n-1)$  en  $a(n-3)$  gelijk te zijn aan het verschil tussen  $a(n-3)$  en  $a(n-5)$ , het verschil tussen  $a(n-2)$  en  $a(n-4)$  gelijk aan het verschil tussen  $a(n-4)$  en  $a(n-6)$ . Daardoor kenmerkt zich dit type van reeksen. De overeenkomstige oplossingsmethode bestaat hierin dat aan  $a(n-1)$  en  $a(n-2)$  de passende verschillen worden toegevoegd om de volgende twee getallen te vormen. De algemene structuur van het programma is zeer eenvoudig : de oplossingsmethoden (acht in aantal) worden in een "compound conditional" ondergebracht (zie fig. 1).

Hoewel de proefpersonen een aantal van dergelijke oplossingsmethoden blijken te bezitten of deze in elk geval kunnen construeren, toch vat een dergelijk type van programma niet alle aspecten die bij het oplossen van een dergelijke opgave een rol spelen. De algemene strategie die door proefpersonen wordt gehanteerd komt neer op een poging om de gegeven reeks te reduceren tot deelreeksen die men a.h.w. automatisch kan verder zetten. Sommige deelreeksen hebben wij allen geautomatiseerd bij het leren van de tafels van vermenigvuldiging, zoals :

7	14	21	28	35	42
---	----	----	----	----	----

Andere reeksen die volstrekt hetzelfde type vertonen zijn helemaal niet geautomatiseerd, neem :

4	11	18	25	32	39
---	----	----	----	----	----

Ons programma behandelt beide reeksen als ekwivalent alhoewel men op basis van retrospectieve gegevens verschillende benaderingen kan verwachten : een "automatisch" voortzetten van de reeks in het eerste geval (met relatief korte reactietijd) en een "+ 7 -regel-georiënteerd" gedrag in het tweede geval (met relatief lange reactietijd). Een meer opvallend verschil komt aan het licht doorheen de vaststelling dat proefpersonen geen onbeperkt lange lijst van methoden kunnen proberen. Na enkele pogingen met diverse type's "hervallen" zij in reeds geprobeerde niet succesrijke methoden. Zij schijnen te vergeten welke analyse's zij reeds hebben uitgetest. Proefpersonen kunnen zich hiervan bewust worden en tot hun ongenoegen vaststellen dat zij na een tijd verweekeld raken in een kringloop, een "loop" van onsuccesrijke methoden. Het uitblijven van succes belet immers niet dat zij steeds weer opnieuw op dezelfde patronen terugkeren, zelfs wanneer ze pogen dit te vermijden.

Het tweede programma dat we voor dit soort gedragingen hebben ontwikkeld poogt de pas vermelde beper-

kingen op het geheugen te combineren met een algemene karakteristiek van de processen waarbinnen de methode of de regel wordt geconstrueerd. Het is namelijk niet zo dat iedereen is voorzien van een geordende reeks onmiddellijk toepasbare methoden voor het oplossen van getallenreeksen. Die methoden worden op basis van een preliminaire analyse ontworpen om vervolgens grondiger te worden uitgetest. Men ziet

3            6            12            24

en men denkt : "meetkundige reeks"; dan ziet men dat na 24 de getallen 44 en 75 volgen en op basis van die gegevens wordt de pas ontdekte methode verworpen en start men een nieuwe preliminaire analyse. Het tweede programma gaat te werk volgens een zeer algemeen analyse-patroon. Een reeks wordt recursief geanalyseerd tot op een niveau waarop een terugkerende sekvens wordt ontdekt. Nemen we de reeks :

1            3            6            12            24            44            75

Daarvoor worden alle verschillen berekend tussen volgenden en voorgaanden, paarsgewijze genomen, wat op zichzelf een nieuwe reeks oplevert :

2            3            6            12            20            31

Er wordt nagegaan of in die reeks een terugkerende sekvens voorkomt. Is dit niet het geval, dan wordt deze reeks als een opgave-reeks behandeld en opnieuw worden alle verschillen berekend :

1            3            6            8            11

Opnieuw wordt gezocht naar een terugkerende sekvens. Ontbreekt een dergelijke sekvens, dan wordt de reeks verschillen nogmaals als een opgave-reeks behandeld. Dit iteratief proces wordt voortgezet tot op het niveau waarop ofwel een terugkerende sekvens wordt ontdekt ofwel slechts één element overblijft ( (n-1) de niveau voor n getallen). In het voorbeeld wordt de regelmaat manifest bij de derde analyse onder de vorm van een herhaling van 2 en 3 :

2            3            2            3

De oplossing omvat, naast het produceren van een volgend getal op basis van de gevonden regelmaat, het toevoegen van de "laatste" elementen die men op elk voorgaand analyse-niveau heeft gevonden :

1	3	6	12	24	44	75	=	119
2	3	6	12	20	31			
	1	3	6	8	11			
		2	3	2	3	<u>2</u>		

De te onthouden elementen (in het voorbeeld : 11, 31 en eventueel het laatste element van de reeks zelf : 75, gesteld dat die niet langer zichtbaar zou blijken) worden opgeborgen in een "stapel" of "push down storage"-element (zie fig. 2).

Een "stapel" is op te vatten als een tijdelijke bewaarplaats, een soort geheugen met als specifiek kenmerk dat de elementen er a.h.w. worden in "opgestapeld" derwijze dat ze een lineair geordende verzameling vormen waarbij slechts aan één kant elementen kunnen worden toegevoegd of weggenomen, nl. aan de "top". Zowel de recursieve analyse als het stapel-element blijken psychologisch relevant. Bij een bepaalde moeilijkheidsgraad vergeten en verwarren de proefpersonen niet alleen de inhoud van de stapel maar ook het aantal elementen in de stapel, een cijfer dat overeenstemt met het aantal niveau's in de recursieve analyse. De taak blijft niet beperkt tot het onthouden van tussenresultaten; ook het niveau dat die resultaten innemen in de recursieve analyse moet worden onthouden. Twee geheugen-factoren spelen aldus een rol : geheugen voor tussenresultaten en geheugen voor lokalisatie van tussenresultaten in het geheel van de analyse. Ten einde het relatief belang van beide factoren na te gaan lijkt het ons aangewezen denktaken te onderzoeken die eenvoudiger zijn en toch een duidelijk hiërarchische structuur vertonen. Met eenvoudiger bedoelen we dan taken waarvoor de proefpersoon over algoritmen beschikt, methoden die hem voor de betrokken opgave een oplossing waarborgen. Opgaven voor hoofdrekenen komen hiervoor in aanmerking.

#### *Hoofdrekenen en computerrekenen.*

Tot nu toe werd de computer slechts vermeld als instrument voor het simuleren van bepaalde aspecten van psychologische processen. Daarbij bleek alleen het programma van belang; hoe dat programma in detail de werking van de machine bepaalt doet immers niets ter zake. Wat we nu ter sprake brengen bevat een analogie tussen bepaalde algemene mentale processen en de werking van bepaalde computersystemen. Strikt genomen gaat het daarbij evenzeer om programma's, nl. de ingebouwde systeem-programma's die de algemene werking van het systeem bepalen : de algemene programma's voor het uitvoeren van de specifieke programma's die de gebruiker er in aanbrengt. Vermits zij door de constructeur zijn ingebouwd vormen die programma's evenwel een essentieel onderdeel van de machine. Het is in termen van die programma's dat we de computer tot model nemen.

Het kan verwondering opwekken dat men poogt in computertermen die menselijke grenzen en beperkingen te reconstrueren die we dank zij diezelfde computer zo ver kunnen overtreffen. Wat kunnen hoofdrekenen en computerrekenen gemeen hebben ? De rekensnelheid van computers is snel maar niet onbeperkt. Eén van de beperkingen op de rekensnelheid ligt in het sekventioneel karakter van de operaties. De computer kan niet een verzameling bewerkingen simultaan uitvoeren (met uitzondering van recente "multi-processing"-systemen). Nu is dit een kenmerk dat ook op hogere mentale processen van toepassing

is. Ook het denken verloopt sekwentioneel, niet parallel (Gregg & Simon, 1967, p. 781).

Een samengestelde opgave als  $(a \times b) + (c \times d)$  is weer te geven als een "liststructure" of boom, een hiërarchie van operaties met als elementen de terminale "mode's" of punten (zie fig. 3). Een computer die  $(a \times b) + (c \times d)$  in een programataal als "input" ontvangt dient deze symbolenreeks vooreerst om te zetten in een sekwentioneel programma. Dit gebeurt via een compileer-proces dat bovenvermelde infixnotatie omzet in een postfixnotatie :

$a b \times c d \times +$

Dit compileer-proces vereist het gebruik van een stapelplaats voor het tijdelijk opbergen van de operatietekens. Voor het uitvoeren van het aldus verkregen sekwentioneel programma is dan nog eens een stapelplaats vereist voor het tijdelijk opbergen van de elementen, waarbij het stapelbeeld als volgt evolueert :

a	b	ab	c	d	cd	(ab+cd)
	a		ab	c	ab	
				ab		

De beschrijving van elementaire opgaven voor hoofdrekenen in termen van "liststructures" laat deze voorkomen als recursieve processen : vermenigvuldigingen worden ontbonden in sommen waarin opnieuw vermenigvuldigingen voorkomen, dit tot op een niveau waarop men de produkten onmiddellijk maken kan. Wanneer we protocollen onderzoeken van proefpersonen die luidop denkend uit het hoofd  $27 \times 13$  berekenen, dan vertoont het plan dat zij daarbij volgen het hiërarchisch patroon van een boom (zie fig. 4). Het gedrag van de proefpersonen verschilt echter hierin van het gedrag van de computer, dat het "compileren" en "uitvoeren" in kleinere segmen gebeurt. Een vermenigvuldiging als  $27 \times 13$  neemt gemiddeld vijftien seconden in beslag. Daarvan verlopen zes seconden vooraleer enig antwoord wordt gerapporteerd. We interpreteren deze eerste fase als het "inlezen" van de opgave en het ontwerpen van een algemeen plan. Vervolgens komt de uitvoering die ongeveer als volgt verloopt (wat luidop wordt gerapporteerd in cursief) :

$$\begin{aligned}
 27 \times 13 &= \underline{27 \times 10} + 27 \times 3 \\
 &= \underline{270} + 27 \times 3 \\
 &= 270 + \underline{20 \times 3} + 7 \times 3 \\
 &= 270 + \underline{60} + 7 \times 3 \\
 &= 270 + 60 + \underline{21} \\
 &= 270 + \underline{81} \\
 &= \underline{351}
 \end{aligned}$$

De niet gerapporteerde segmenten, zo veronderstellen we, worden in een stapelplaats opgeslagen. Daarin bevinden zich aldus zowel de tussenresultaten als de nog gedeeltelijk te "compileren" segmenten van het programma. Strikt genomen moeten we zelfs twee stapels veronderstellen : een stapel van tussenresultaten en een stapel van de nog uit te voeren bewerkingen. De strategie die de proefpersonen volgen toont aan

dat zij pogen het aantal elementen in beide stapels minimaal te houden. In termen van de hiërarchische beschrijving komt dit hierop neer dat zij het aantal punten in de boomstructuur minimaal houden. Bij het maken van het plan ontwikkelen zij niet vooraf de volledige boom om dan over te gaan tot de uitvoering. Vooreerst wordt de vertakking links ontwikkeld en uitgewerkt en door een tussenresultaat vervangen, dan wordt de vertakking rechts ontwikkeld.

De aspecten van het geheugen die relevant bleken bij het oplossen van getallenreeksen komen dus ook hier aan bod : geheugen voor tussenresultaten en geheugen voor plan-aspecten. Hier kunnen we echter duidelijker de vergeet-fenomenen bestuderen. Worden tussenresultaten meer vergeten dan plansegmenten? Welke tussenresultaten en welke plansegmenten worden vergeten? Voorlopig laten onze beperkte gegevens niet toe deze vragen te beantwoorden. Bij gecompliceerde opgaven worden zowel tussenresultaten als plansegmenten vergeten : voornamelijk de bodemelementen van beide stapels. Interferentie tussen beide stapels laat daarbij twijfel oprijzen aangaande de psychologische realiteit van dergelijke geheugenelementen. Deze blijken niet onderling onafhankelijk. Sommige proefpersonen schijnen op bepaalde momenten aan te voelen dat een element in een van de stapels op het punt staat vergeten te worden en zij voorkomen dit door het element mentaal te "herhalen". Dit "herhalen" interfereert echter met het onthouden van elementen in de andere stapel. Iets herhalen om vergeten te voorkomen van een element in de stapel "tussenresultaten" kan het vergeten veroorzaken van een element in de stapel van de nog uit te werken opgaven. Een betere voorstelling gaat dan ook in de richting van een combinatie van de twee stapелеlementen, a.h.w. horizontaal met de toppen naar elkaar toe geplaatst aan weerszijden van de "reken-eenheid", waarbij het geheel globaal genomen een beperkte capaciteit heeft en elk geheugenelement kan worden geëxpandeerd, evenwel ten koste van de inhoud van het andere element. Die globale capaciteit kan dan overeenstemmen met de capaciteit van het onmiddellijk geheugen en de "span of attention".

#### *Hiërarchische organisatie en aandacht*

Sommige psychologen hebben computersimulatie opgevat als vrij beperkt in waarde gezien het "single minded, undistractable, and unemotional" karakter van simulatieprogramma's (Neisser, 1967, p. 9). Schaakprogramma's spelen slechts schaak. Zij produceren geen woede of ontgoocheling wanneer zij het spel verliezen, geen gevoelens van triomf wanneer zij een overwinning boeken. De door ons vermelde programma's lossen slechts bepaalde soorten getallenreeksen op. Kunnen zij zich op een zeer beperkt domein intelligent tonen, voor intelligentieproeven van andere aard blijven zij onbewogen. Simulatieprogramma's zouden dan ook slechts van betekenis zijn voor datgene wat zij simuleren tenzij zij een of andere belangrijke karakteristiek gemeen hebben die een meer algemeen aspect van menselijke gedragingen zou reflecteren. Is er zo'n karakteristiek ?

Zowel bij het oplossen van getallenreeksen als bij het hoofdrekenen bleek een hiërarchische structurering psychologisch relevant. Hiërarchische structuren komen onder de vorm van "liststructures" en bomen



frekwent voor in simulaties van cognitieve processen en "artificial intelligene"-studies. Het EPAM-programma (Elementary Perceiver And Memorizer) van Feigenbaum en Simon (1962) heeft discriminatienetten onder de vorm van bomen. Diverse stappen in een logische bewijsvoering, "generaties" van mogelijke zetten vanuit een bepaalde positie in spelen zoals dammen en schaken worden in de overeenkomstige programma's eveneens onder de vorm van bomen weergegeven. Uit programma's voor "problem solving" ontwikkelt zich trouwens een algemene theorie voor het "doorlopen van bomen" (tree-search) waarbinnen diverse strategieën voor het expanderen van dergelijke grafen kunnen worden bepaald (Nilsson, N.J., 1969). Ook de generatieve linguïstiek van Chomsky (1957) waarin boomstructuren als "phrase-markers" opduiken, heeft tot de populariteit ervan bijgedragen. Yngve (1961) heeft op basis van die "P-marker"-grammatica en Miller's "magical number 7" (Miller, G.A., 1956) een model voor taalproductie voorgesteld waarin het ontwikkelen van bomen beperkingen vertoont die nauw verwant zijn met de restricties die wij menen vast te stellen bij hoofdrekken. Vermelde toepassingen hebben echter betrekking op hogere mentale processen. Kunnen we de hiërarchische beschrijving wel op elk gedrag toepassen? Ook in die richting werden voorstellen gedaan, zij het lang geleden, o.a. door Holt (1915) en Perry (1918, 1921). Hoe beschrijft men een man die een huis zoekt te huren? De man staat op van zijn stoel, trekt zijn jas aan, zet een hoed op, gaat naar een tramhalte, wacht, stapt op de aankomende tram, betaalt een kaartje, stapt x haltes verder af, gaat een leegstaand huis binnen, loopt er in rond, gaat weer buiten, stapt naar een verhuurkantoor ... Dat alles hangt niet aaneen als schakels van een ketting, het vormt een "zinnig" geheel, als een "zin" moet het worden begrepen. De zin is dat die man een huis zoekt te huren. Het "hoed opzetten" hoort bij het "jas aantrekken" zoals een adjectief bij een substantief hoort. "Zich aankleden" kan worden geëxpandeerd in "sjaal omdoen + jas aantrekken + hoed opzetten". Het vormt een hiërarchische structuur die, onder de controle van een hogere "mode" of punt, met vele andere gedragingen kan worden gecombineerd. Kunnen we het gedrag beschrijven in termen van de expansie van een boom dan kunnen we het fenomeen aandacht beschrijven in termen van het aantal punten dat tijdelijk moet worden onthouden. Stellen we daarbij voorop (in overeenstemming met Yngve's vooropstelling) dat de boom wordt ontwikkeld of doorlopen van boven naar onder (top down) en van links naar rechts, dan kunnen we aangeven hoe het komt dat we in bepaalde situaties de draad kwijt raken, dat bepaalde dingen aan onze aandacht ontsnappen. Te veel vertakkingen naar links, d.i. regressieve vertakking, houden in dat een belangrijk onderdeel van de taak zal worden vergeten: de rechtervertakking op het hoogste niveau.

Aandacht is veelal geïnterpreteerd geworden als het produkt van een voortdurende selectie van de zintuiglijke prikkels waartegenover wij reageren. Er valt op elk moment ontzettend veel te zien en te horen; onze aandacht zou instaan voor de keuze van wat we effectief bekijken en beluisteren. Teoretisch leidt dit tot een soort paradoxale situatie: immers, om de keuze te maken moet minstens een gedeeltelijke analyse van de prikkels worden doorgevoerd. Om in staat te zijn te kiezen moeten we weten wat de omgeving ons biedt. Theorieën die de aandacht vanuit de waarneming benaderen hebben hiermee steeds moeilijkheden. Soms wordt de opgave zo complex dat het er naar uitziet alsof de mens, in analogie met bepaalde "time-sharing"-systemen, het grootste gedeelte van zijn tijd wijdt aan keuze-problemen aangaande "inputs" en niet aan die "inputs" zelf. Aandacht is evenwel niet tot de "input"-zijde beperkt. Ook aan de "output"

of motorische zijde kan men het aandachtsfenomeen terugvinden, zij het dan onder de naam van "set". De starthouding van een hardloper levert het beeld van een organisme dat volledig is ingesteld op "lopen". Er is een tijdelijke selectiviteit ten aanzien van mogelijke bewegingen. Deze selectiviteit moet echter ook ten aanzien van mentale entiteiten en interne processen worden uitgebreid, ideeën die ons bij het denken voor de geest komen ondergaan immers een zelfde selectie. De aandachtsfunctie stemt gedeeltelijk overeen met de controlefunctie in een computer en is een mechanisme

- a) voor het controleren van het planonderdeel dat in uitvoering is en
- b) het bepalen van het volgend plansegment.

Op elk moment omvat de aandacht in de grafische weergave van het plan de onmiddellijke context van punten omheen dat punt van de boom dat de activiteit van dat moment bepaalt. Wanneer we een primate van het handelen op de waarneming vooropstellen kunnen de moeilijkheden met betrekking tot de selectie van "inputs" worden opgeruimd. Het is een plan, een programma voor externe of interne activiteit, dat ons "instelt" op bepaalde aspecten van onze omgeving en bepaalde sectoren van ons geheugen. Onze activiteiten zelf vormen de filters doorheen dewelke "inputs" worden geselecteerd.

\* Een gedeelte van de onderzoeken werd voorbereid en uitgewerkt tijdens een verblijf aan het Center for Cognitive Studies, Harvard University, Cambridge, U.S.A., 1966-1967. Dit verblijf werd mogelijk gemaakt door een Frank Boas-scholarship en de United States Educational Foundation in Belgium. Vermelde computerprogramma's werden gerealiseerd op het M.I.T.'s Project MAC time-sharing systeem met de hulp van Dr. Philip J. Stone.

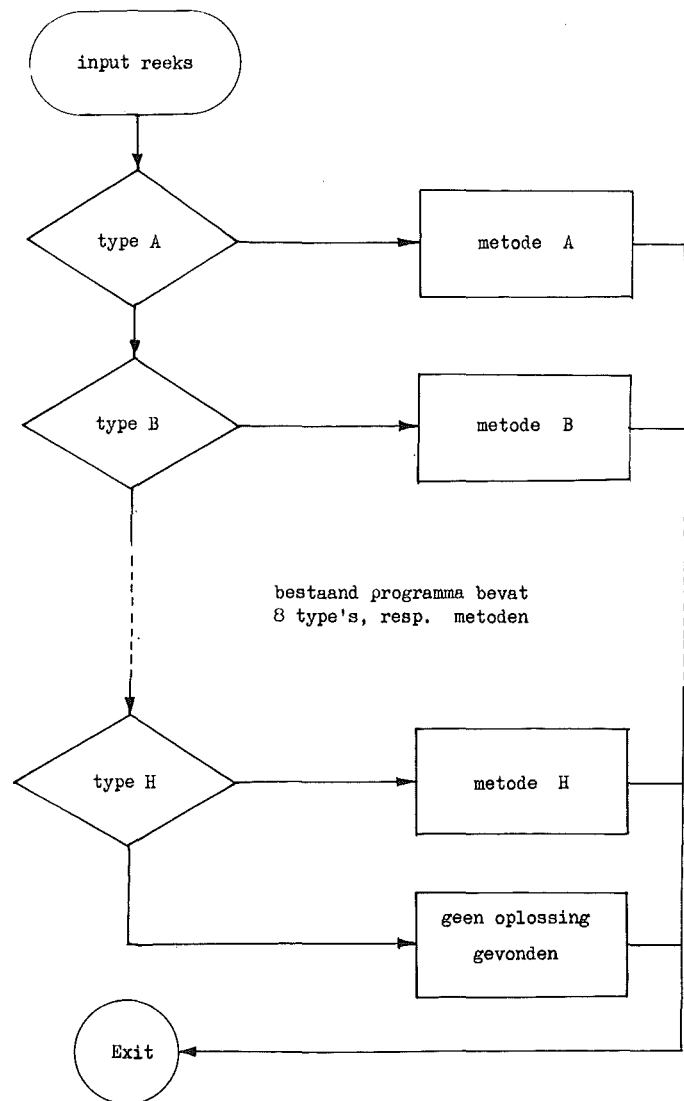


Fig. 1. Programma voor het oplossen van getallenreeksen onder de vorm van een "compound conditional"

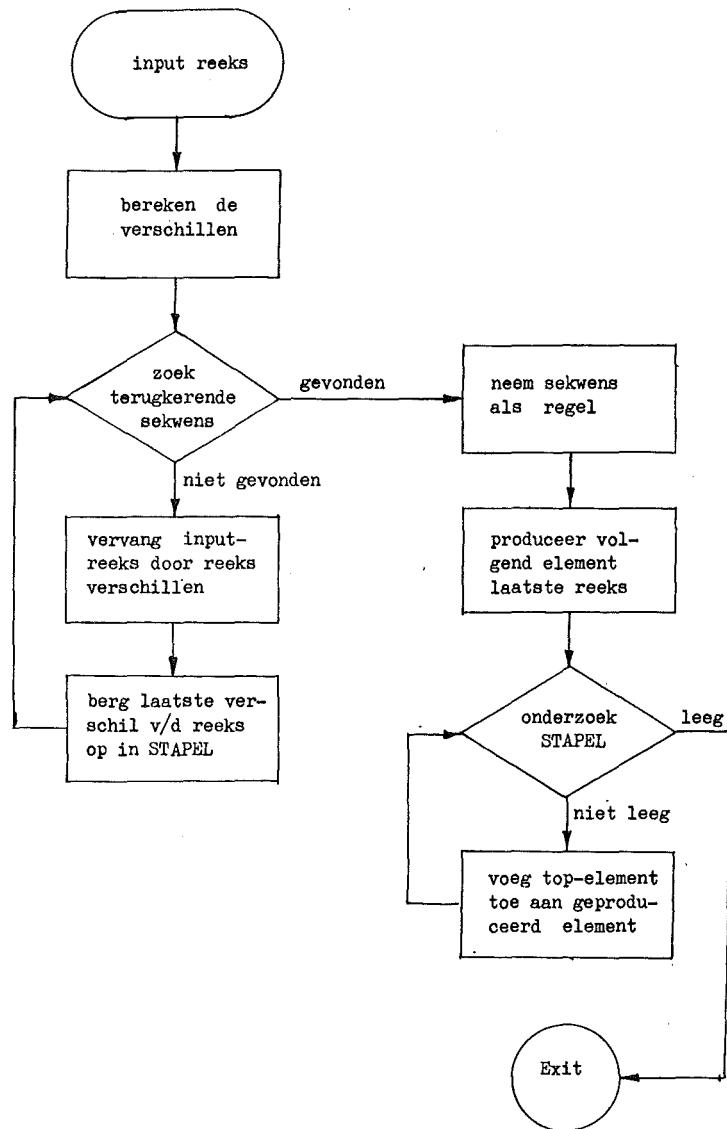


Fig. 2. Recursief programma voor het oplossen van getallenreeksen

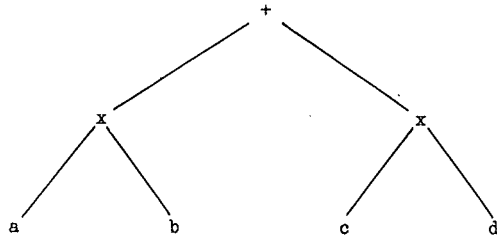


Fig. 3. Boom-representatie van de som  $(a \times b) + (c \times d)$

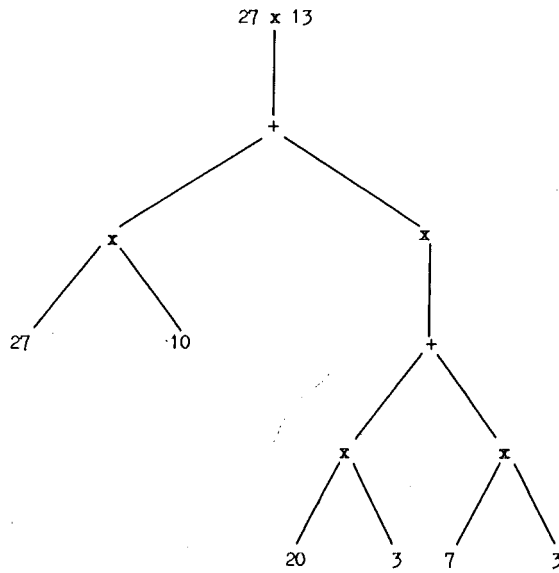


Fig. 4. Hiërarchisch plan voor het "uit het hoofd" berekenen van de opgave  $27 \times 13$

## REFERENTIES

- Chomsky, N. *Syntactic Structures*. The Hague, Mouton & Co, 1957.
- Feigenbaum, E. A. Information processing and memory. In : Norman, D. A. (ed.) 1970, 451-468.
- Feigenbaum, E. A. and Simon, H. A. A theory of the serial position effect. *British Journal of Psychology*, 1962, 53, 307-320.
- Frijda, N. H. La simulation de la mémoire. In : *La Mémoire*, Symposium de l'Association de psychologie scientifique de la langue française, Genève 1962. Paris, P.U.F., 1970.
- Gregg, L. W. and Simon, H. A. An information-processing explanation of one-trial and incremental learning. *Journal of Verbal Learning and Verbal behavior*, 1967, 6, 780-787.
- Holt, E. Response and cognition. 1915. In : Holt, E. 1916.
- Holt, E. *The Freudian Wish and its Place in Ethics*. New York, Holt, 1916.
- Laughery, K. R. and Gregg, L. W. Simulation of human problem-solving behavior. *Psychometrika*, 1962, 27, 265-282.
- McLeod, J.P. (ed.) *Simulation*, the dynamic modeling of ideas and systems with computers. New York, McGraw-Hill, 1968.
- Miller, G.A. The magical number seven, plus or minus two; some limits on our capacity for processing information. *Psychological Review*, 1956, 63, 81-96.
- Neisser, U. *Cognitive Psychology*. New York, Appleton-Century-Crofts, 1967.
- Newell, A. and Simon, H. A. Computer simulation of cognitive processes. In : Luce, R. D., Bush, R. R., Galanter, E. (eds.) *Handbook of Mathematical Psychology*, vol. 1, New York, Wiley, 1963, 361-428.
- Milsson, N. J. Searching problem-solving and game-playing trees for minimal cost solutions. In : Morrell, A. (ed.) *Information Processing 68*, Proceedings of IFIP Congress 1968, vol. 2. Amsterdam, North-Holland 1969, 1556-1562.
- Norman, D. A. (ed.) *Models of Human Memory*. New York, Academic Press, 1970.

Oettinger, A. G. *Run, Computer, Run*. The mythology of educational innovation. Cambridge, Mass., Harvard University Press, 1969.

Perry, R. B. Docility and purposiveness. *Psychological Review*, 1918, 25, 1-20.

Perry, R. B. A behavioristic view of purpose. *Journal of Philosophy*, 1921, 18, 85-105.

Simon, H. A. and Kotovski, K. Human acquisition of concepts for sequential patterns. *Psychological Review*, 1963, 70, 534-546.

Yngve, V. The depth hypothesis, structure of language and its mathematical aspects. *Proceedings of Symposia in Applied Mathematics*, American Mathematical Society, vol. XII, 1961.

### *SUMMARY*

The studies reported form part of a project aimed at understanding the rôle of memory and attention in human behavior. Two programs for the completion of number series are presented. Such series are common in tests measuring intelligence. They consist of sequences of numbers governed by some well defined rule. The programs are discussed in terms of their value as simulations of cognitive processes involved in similar tasks. The first program includes a simple list of methods, the second one is recursive and rule-oriented. Compared with the performance of subjects, both programs show the necessity for built in memory restrictions. Subjects' memory is limited with respect to the number of methods tried as well as with respect to intermediate results and level of analysis in recursive reconstruction. In order to investigate those restrictions on memory, mental multiplication is studied in some detail; mental arithmetic exhibits more clearly the hierarchical structure associated with recursive reconstruction. The decomposition of complex mental problems into manageable units is compared with sequentialisation in computers. Planning and control of multiplications and also of other behaviors seem to involve the expansion of tree-structures having a limited number of nodes simultaneously apprehended. That number corresponds to the "span of attention", attention being a mental "function" comparable to the control unit in a computer.