

Computer gestuurd denken

Training in het hanteren van algoritmische en heuristische probleemoplossingsmethoden*

L. DE LEEUW

Psychologisch Research Laboratorium, Subfakulteit Psychologie, V.U., Amsterdam

Samenvatting

Allereerst wordt een toelichting op het begrip 'leren denken' en het onderscheid tussen algoritmische en heuristische oplossingsmethoden gegeven. Er worden relaties gelegd met de theorie van de Russische psychologen Landa en Kuljutkin, met het werk van Selz en zijn medewerkers en met de Amerikaanse 'discovery' vs regel-leren methode. Vervolgens wordt ingegaan op de methodologische aspecten van onderzoek naar 'leren denken' zoals die door Wittrock en Cronbach naar voren zijn gebracht. De eigen onderzoekopzet wordt aan de hand van de opgesomde punten van kritiek toegelicht, waarbij ook wordt uiteengezet hoe aan het door Cronbach gepropageerde zoeken naar een vijfvoudige interactie t.w. tussen leerstof, type instructie, leertijd, type leerling en leerresultaat in het onderzoek is vorm gegeven. Voorts wordt duidelijk gemaakt dat een individuele onderwijsmethode waarbij de leerling actief bezig is, in eigen tempo kan werken, een bepaalde mate van eigen keuze heeft op de gang door het leerprogramma en waarbij de feedback afhankelijk is van de leergeschiedenis alleen mogelijk is met behulp van computer-gestuurd onderwijs.

De twee gebieden waarin de problemen liggen die de leerlingen moeten leren oplossen t.w. syllogistisch redeneren (deductief redeneren) en getalrijen extrapoleren (inductief redeneren) alsmede de beide klassebreedten die binnen elk gebied worden onderscheiden worden nader gedefinieerd. Naast de hoofdfactoren van het design (2×2) t.w. algoritmisch-heuristisch en klassebreedte (heterogeniteit van probleem-typen) wordt de invloed van een aantal leerlingkenmerken onderzocht. Er wordt een aantal criteria (afhankelijke variabelen) gedefinieerd. M.b.t. de onderwijsprogramma's wordt alleen op de getalrijen programma's ingegaan.

Na het vermelden van een aantal door Landa geformuleerde uitgangspunten voor het aanleren van

algoritmische oplossingsmethoden en het noemen van een aantal punten van kritiek op zijn werkwijze wordt de structuur en inhoud van het algoritmisch leerprogramma 'getalrijen extrapoleren' uiteengezet. Daarbij komt onder meer aan de orde: het inoefenen van de verschillende subalgoritmen teneinde de operaties voor alle leerlingen elementair te maken en de wijze van 'vanishing' (geleidelijke uitwissing) van het algoritmisch schemegebruik.

Na een beschouwing van de heuristische oplossingsmethode wordt uiteengezet hoe de leerling tot een wendbare handelingsstructuur wordt gebracht. De structuur van het heuristisch leerprogramma komt aan de orde. Het gebruik van standaard teaching-logics (Lehralgorithmen) wordt benadrukt. Het gebruik van multiconditionele feedback en prompting alsmede het gebruik van de computer en randapparatuur hierbij wordt uiteengezet.

Het artikel wordt besloten met een toelichting op de keuze van de transfer-tests; hierbij komt het 'Structure of Intellect'-model van Guilford als heuristiek ter sprake.

Buiten een themanummer als het onderhavige zou dit artikel gezien de titel moeten beginnen met een uitvoerige theoretische uiteenzetting over de begrippen algoritmische en heuristische oplossingsmethoden.

Nu deze begrippen in het artikel van prof. Van Parreren uitvoerig aan de orde komen kan de beschikbare ruimte besteed worden aan opzet en uitwerking van het onderzoek.

Wel wil ik enige kanttekeningen maken bij de begrippen 'leren denken' en de in de Russische literatuur behandelde begrippen 'algoritmische en heuristische oplossingsmethoden'.

Onderzoek op het gebied van het leren denken is bepaald niet nieuw. We hoeven slechts namen als Selz (1935) en Kohnstamm (1948) te noemen om dat duidelijk te maken. Met name Selz rapporteert zeer frappante resultaten.

Leren denken kan plaatsvinden door de *oplossingsmethode* voor bepaalde probleemtypen aan te

* Dit onderzoek maakt deel uit van het S.V.O. project 0113: Computer Assisted Instruction; Programmering van Responsive Environments.

leren. Het werken volgens een oplossingsmethode impliceert dat het subject een schematische voorstelling heeft van het verloop van het denkproces. Dit schema geeft de volgorde, de structuur van de stappen aan die het denkende subject moet maken om de oplossing te bereiken.

Zo'n schema moet adaptief zijn, zodat aanpassing bij de tussentijdse uitkomsten van het denkproces mogelijk is. Van Parreren (1969) spreekt van cognitieve schema's (handelingsstructuren). In de terminologie van Van Parreren (1969) hebben wij hier te doen met een bepaald type cognitief schema, dat gezien kan worden als een oplossingsmethode voor taken van dezelfde soort. Deze oplossingsmethoden vormen een hiërarchie van specifiek naar algemeen.

Als men leerlingen wil leren denken, dan kan men zoals gezegd dat doen door hen voor bepaalde probleemklassen de oplossingsmethode aan te leren. De vraag rijst dan welke mate van geprestructureerdheid deze oplossingsmethode dient te hebben. Men kan een handelingsschema aanleren waarin voor de betreffende klasse van problemen de stappen die tot de oplossing leiden zeer expliciet en in de juiste volgorde worden voorgeschreven. Men heeft dan een *algoritme*: een serie operaties en toetsingen die mits juist uitgevoerd *oplossingsgarantie* bieden. Men kan de leerling ook minder schematisch te werk leren gaan, hem zelf bepaalde relaties laten ontdekken, waarbij de leerling moet leren om als bepaalde relaties niet tot de oplossing leiden andere relaties te zien en op hun explicatieve waarde (zie hierna) te leren onderzoeken. De leerling moet zichzelf dan aanwijzingen en hints leren verschaffen om vastlopen in de probleemsituatie te voorkomen. Men heeft dan een *heuristische* oplossingsmethode.

Hierbij dient te worden aangetekend dat er hier niet van een dichotomie maar veeleer van een continuum sprake is. Ook Landa (1971) onderscheidt algoritmische, semi-algoritmische, semi-heuristische en heuristische voorschriften.

Het begrip heuristische methode verdient enige nadere toelichting. Kuljutkin (1970) stelt dat het in een probleemsituatie gaat om het op het spoor komen van de explicatieve relaties; dit zijn de relaties tussen elementen van de probleemsituatie die voor de oplossing van belang zijn. Zodra de explicatieve relaties bekend zijn volgt dan het implicatief proces (omdat dit soort relatie aanwezig is geldt dat...).

Het onderscheid tussen algoritmische en heuristische oplossingsmethoden wordt volgens Kuljutkin gevormd door het al dan niet *bekend* zijn van de

explicatieve relaties die in de probleemsituatie aanwezig zijn. Dit onderscheid behoeft naar mijn mening verheldering. Een algoritme is opgebouwd uit sequenties van toetsingen en operaties; in termen van Miller, Galanter en Pribram (1960) vormt een algoritme een TOTE-systeem. De toetsingsstappen kunnen bestaan uit de opdracht na te gaan of bepaalde explicatieve relaties aanwezig zijn. Voordat de probleemoplosser deze toets heeft uitgevoerd weet hij nog niet of de betreffende explicatieve relatie in deze probleemsituatie aanwezig is; wel is hij, doordat hij het algoritme kent, bekend met de mogelijkheid van het bestaan van dit soort relatie in dit type probleemsituaties. Het gaat hier dus niet om het ontdekken van een nieuw onbekend soort relatie maar om het vaststellen of een bekend soort relatie aanwezig is.

Als de probleemsituatie heuristisch wordt opgelost kan de betreffende explicatieve relatie aan de probleemoplosser onbekend zijn; hij moet dan zelf het bestaan van dit soort relatie ontdekken. De formulering zoals die voorkomt in Van Parreren en Van Loon-Vervoorn (1975, blz. 84) – het betreft een samenvatting van het boek van Kuljutkin – luidt: '... zijn de benodigde explicatieve relaties onbekend, dan kan men spreken van een heuristisch proces'.

Deze formulering laat open of met 'onbekend' bedoeld wordt of de betreffende explicatieve relatie, nog niet voor dit probleem is vastgesteld of dat de probleemoplosser de potentiële relatie voordat hij de oplossing vindt niet kent.

Als 'onbekend' in eerstgenoemde zin (nog niet voor dit geval vastgesteld) moet worden opgevat, dan vormt het bekend of onbekend zijn van de betreffende explicatieve relatie(s) geen goed criterium voor het onderscheid tussen algoritmische en heuristische processen. Dit zal nader worden toegelicht als in het hiernavolgende concrete voorbeelden van algoritmische en heuristische oplossingsmethoden behandeld worden.

Selz (1935) heeft beschreven hoe Sand, een leerling van hem, zwakbegaafde leerlingen van het basisonderwijs getalreeksen heeft leren oplossen door ze de specifieke gedragswijzen ('specifieke Verhaltungsweisen') die daarvoor nodig waren aan te leren door systematisch opgebouwde training. Dit is een uitwerking van Selz' gedachte van de synthetische opbouw van intelligentie-prestaties. Sand werkte klassikaal en individueel volgens het principe der 'kleinstmögliche Hilfe', een ontdekkingsproces dus met een geringe mate van sturing ('guidance, cueing, prompting'). In modernere termen gesteld

kunnen we zeggen dat een heuristische oplossingsmethode werd aangeleerd.

De succesvolle pogingen van Sand namen 16 klassikale lessen in beslag plus een niet te achterhalen aantal individuele lessen. De vraag rijst of dit via een andere aanpak wellicht sneller kan met evenveel en mogelijk zelfs meer kans op succes.

Naast de heuristische aanpak van Sand (die overigens nadere operationalisatie behoeft) zou een algoritmische aanpak gevolgd kunnen worden, waarbij leerlingen systematisch leren nagaan of bepaalde principes in een rij aanwezig zijn. Ze maken dan in het leerprogramma kennis met de verschillende principes (soorten regelmatigigheden) en leren vervolgens zichzelf systematisch afvragen of bepaalde principes, bepaalde explicatieve relaties in de aan te vullen getalrij aanwezig zijn. Belangrijk criterium bij het vaststellen van het differentieel effect van de beide soorten oplossingsmethoden is de *transfer* die optreedt naar niet getrainde typen. Het is niet ondenkbaar dat, in termen van Van Parreren, het ene soort aangeleerde handelingsstructuur (oplossingsmethode) een grotere mate van *wendbaarheid* zal geven dan het andere; zelfs negatieve transfer is niet uitgesloten.

Methodologische aspecten van onderzoek naar 'leren denken'

In de Russische onderzoeken naar leren denken en het leren van probleemoplossingsmethoden wordt aan de methodologische kant van het onderzoek betrekkelijk weinig aandacht besteed.

In het Amerikaanse onderzoek dat gericht is op de vergelijking van discovery-learning en expositorisch leren – onderwijsprocedures die sterk verwant zijn met de wijze waarop in ons onderzoek resp. heuristische en algoritmische oplossingsmethoden worden aangeleerd¹ – wordt aan de methodologische aspecten meer aandacht besteed.

Wittrock (1966) somt in 'Learning by Discovery', een verslag van een congres geheel gewijd aan 'discovery learning', een aantal methodologische bezwaren op tegen onderzoek waarin 'discovery learning' en expositorisch leren (regel leren) op effect zijn vergeleken.

1. De *niet repliceerbaarheid* van de 'treatments' (onderwijsprocedures). In veel experimentele studies zijn de onderwijsprocedures niet operationeel gedefinieerd. Het zijn complexe en langdurige sequenties van stimuli die vaak op een aantal manieren van andere onderwijsprocedures verschillen.

2. Het ontbreken van *random toewijzing* van leerlingen aan de onderwijsprogramma-versies. Vaak wordt gewerkt met *intacte groepen* (schoolklassen). Daarom zijn veel van de studies die experimenten heten geen echte experimenten.
3. Inadequate *statistische analyse*. Soms helemaal geen, soms een zeer summiere vermelding hiervan.
4. *Voorbarige extrapolatie* van de resultaten. Zelden worden binnen één studie verschillende soorten leerstof en verschillende typen leerlingen bestudeerd. Wel hebben er extrapolaties plaats naar populaties en leerstofgebieden die niet in het onderzoek zijn opgenomen.

Naast deze methodologische kritiek noemt Wittrock nog een aantal steekhoudende semantische en conceptuele bezwaren.

1. Onderwijsprocedures worden vaak benoemd in termen van de *resultaten* die ze geacht worden te produceren. Zo zijn 'rote' en 'discovery' twee gangbare benamingen. Er zijn minder emotioneel geladen meer neutrale namen nodig om deze procedures te beschrijven ('discovery' suggereert iets hogers, 'rote learning' iets lagere).
2. Het ontbreekt aan *operationele definities* van de term 'discovery'. Soms wordt de verbalisatie van zeer algemene en vage aanwijzingen (cues) ook een 'discovery'-procedure genoemd. Vaak ook wordt van 'discovery' gesproken als de leerkracht niet regels en generalisaties maar een variëteit van specifieke problemen verwoordt. 'Discovery' wordt soms als *middel* beschreven, soms als *doel* op zichzelf. M.a.w. 'discovery' wordt niet alleen gezien als middel om de structuur van een stuk leerstof te leren, maar ook als middel om te leren probleemoplossen. Deze twee doeleinden moeten strict gescheiden worden gehouden en operationeel worden gedefinieerd.
4. De onderzoekers van 'discovery learning' kenmerken zich door hun *depreciatie van verbaal leren*. De leerkracht moet zo min mogelijk zeggen, zo min mogelijk aanwijzingen geven. Dit is een misvatting. Er is veel te zeggen voor het zgn. geleide, gestuurde ontdekken ('guided discovery').
5. De mogelijkheid om de leerling zijn *eigen tempo* alsmede zijn *eigen volgorde* van stimuli te laten bepalen krijgt te weinig aandacht.
6. Er worden zeer sterk wisselende afhankelijke variabelen (*criteria*) gehanteerd in de verschillende studies. Soms is het criterium transfer, soms retentie, soms het vermogen problemen op te lossen.

Cronbach (1966) sluit zich in een ander hoofdstuk van het betreffende congresverslag op een aantal punten bij hem aan, maar legt de accenten wat anders. Hij stelt de vraag of 'discovery learning' een betere methode is dan 'didactic teaching'².

Zijn antwoord is dat deze vraag in zijn algemeenheid niet is te beantwoorden. Het zoeken moet zijn naar beperkte generalisaties in de vorm:

Met leerstof van deze aard
produceert discovery learning van dit type
bij deze mate van oefening
dit leerresultaat
van leerlingen met deze persoonlijkheidsken-
merken.

Cronbach vat hier heel kernachtig een groot aantal van de door Wittrock genoemde punten samen. Alle door hem genoemde aspecten hebben betrekking op de externe validiteit (Campbell en Stanley, 1963). Door te vragen naar specificatie van de leerstof waarover een uitspraak wordt gedaan wordt gewaarschuwd voor voorbarige extrapolatie van resultaten. Het doen van uitspraken over specifieke typen 'discovery learning' impliceert nadruk op de operationele definitie van de 'discovery' onderwijsprocedure, waardoor deze procedure ook replicerbaar wordt. Hiermee wordt o.m. de mate van sturing ('guidance') die aan het ontdekkingsproces wordt verleend vastgelegd.

Ook de duur van het leerproces is een variabele. Bovendien valt aan te nemen dat niet voor alle leerlingen de ene manier van kennisverwerven beter is dan de andere, maar dat dit afhankelijk is van leeftijd, persoonlijkheidskenmerken en leerstijl van de leerling. Dit loopt vooruit op het later door Cronbach gepropageerde 'aptitude-treatment' interactie onderzoek.

Als maten van het leerresultaat noemt Cronbach o.m.: toepassing (transfer), retentie, het vermogen om nieuwe regels te ontdekken, interesse.

Hij concludeert dat er complexere experimenten nodig zijn waarin *vijfvoudige* interacties worden onderzocht t.w.: leerstof met type instructie, met leertijd, met type leerling, met resultaat.

Onderzoek-opzet

Hoe moet een onderzoek waarin algoritmische en heuristische oplossingsmethoden worden vergeleken worden opgezet, zodanig dat de genoemde punten van aanbeveling en kritiek daarin mede zijn verwerkt? Een dergelijk onderzoek moet door het volgende gekenmerkt zijn.

1. Er moet met meer dan één soort leerstof worden gewerkt.
2. Er moet met meer dan één type leerling worden gewerkt.
3. De leerlingen moeten kunnen werken in eigen tempo en moeten hun eigen volgorde van stimuli te verwerken krijgen.
4. De treatments moeten operationeel gedefinieerd zijn zodat ze replicerbaar zijn.
5. Er moeten random toewijzing van leerlingen aan de treatments plaats hebben.
6. Omdat de leerling de oplossingsmethode tenslotte zelfstandig moet hanteren, moet hij niet receptief maar actief en in de heuristische versie explorerend bezig kunnen zijn. De onderwijzende instantie moet op deze acties adequaat en sturend reageren.
7. Er moeten meerdere criteria voor het leerresultaat (afhankelijke variabelen) worden vastgesteld.
8. Er moeten relevante leerlingkenmerken die interacties met de onderwijsprocedures kunnen geven worden vastgesteld.

Deze punten zijn voor ons onderzoek door de volgende onderzoekopzet gerealiseerd:

ad 1 Er zijn twee leerstofgebieden (beter: denkstofgebieden) gekozen. Deze liggen op het terrein van het inductieve en het deductieve redeneren t.w. resp. het extrapoleren van getalrijen en syllogismen-redeneringen.

De leerstof is op een tweede manier gevarieerd door binnen elk der twee denkstofgebieden de breedte van de klasse van problemen variabel te maken. In de ene klassebreedte worden meer typen opgaven aangeleerd dan in de andere; de klassen verschillen dan in de mate van homogeniteit/heterogeniteit.

De overweging bij het toevoegen van de klassebreedte-factor is geweest dat het niet ondenkbaar is, dat bij het leren oplossen van een beperkte, vrij homogene klasse van problemen een meer specifieke oplossingsmethode, waarin de leerling stap voor stap naar de oplossing wordt geleid (algoritmische methode), meer effect heeft dan een meer algemene methode, waarin wel aanwijzingen en hints worden gegeven, maar waarbij toch meer aan de inventiviteit van de leerling wordt overgelaten (heuristische methode).

Bij een meer heterogene probleemklasse, waarin meer typen problemen voorkomen, zouden de verbanden wel eens anders kunnen liggen. Een eventuele interactie (Bracht en Glass, 1968) tussen type oplossingsmethode en klassebreedte zou tot stand kunnen komen doordat voor een heterogene klasse van problemen het algoritme erg ingewikkeld kan

worden, zodat dit *moelijk leerbaar* is of, als het geleerd wordt, snel *vergeten* wordt. Het laatste punt is een reden om naast zgn. post-tests ook retentie-tests af te nemen (zie ad 7).

Binnen elk denkstofgebied en elke klassebreedte wordt het probleemoplossen zowel middels een algoritmische als middels een heuristische oplossingsmethode getraind. Uiteraard wordt per klassebreedte en per soort oplossingsmethode met andere groepen leerlingen gewerkt. Per denkstofgebied nemen zodoende vier verschillende (vergelijkbare) groepen leerlingen deel aan het onderzoek.

ad 2 De denkstofgebieden zijn zodanig gekozen, dat leerlingen van verschillende leeftijden en begaafdheden hiermee kunnen leren werken. In eerste instantie zijn de problemen op het gebied van het inductieve redeneren (getalrijen) bestemd voor leerlingen van de vijfde en zesde klas van de basisschool. De syllogismen-redeneringen (deductief redeneren) voor leerlingen van de vierde en vijfde klas van de middelbare school.

ad 3 Door gebruik te maken van computer-gestuurd onderwijs (CAI = Computer Assisted Instruction) is het mogelijk leerlingen in eigen tempo te laten werken, terwijl iedere leerling zijn eigen pad door de leerstof volgt. Bovendien kan – volgens hierna te specificeren procedures – de leerling in een bepaalde mate bewust invloed uitoefenen op de informatie die hem wordt aangeboden³. Het onderwijsleerproces wordt gestuurd door het computerprogramma. Hierin kunnen conditionele opdrachten worden opgenomen, zodat bij het aanbieden van nieuwe informatie en feedback rekening kan worden gehouden met al eerder door de leerling ondernomen oplossingspogingen of door hem gemaakte fouten van een bepaald type.

ad 4 Doordat zowel de 'courseware' (leerstof-inhoud) als de 'software' (het sturingsprogramma dat de leerling op grond van zijn reacties naar nieuwe leerstofonderdelen stuurt) zijn vastgelegd zijn de treatments operationeel gedefinieerd en is het experiment replicerbaar. Bovendien worden alle acties en reacties van de leerling geregistreerd.

ad 5 Computer-gestuurd onderwijs is individueel, zodat de leerlingen op grond van, tevoren gemeten, relevant geachte indelingscriteria aan de verschillende treatments kunnen toegewezen.

ad 6 Ideaal is de leerling niet te laten kiezen uit gegeven antwoordalternatieven (zoals in vertakte geprogrammeerde instructie wel het geval is), maar de leerlingen de antwoorden zelf te laten construeren.

Bij gebruik van computer-gestuurd onderwijs is het mogelijk leerlingen antwoorden te laten typen op een zgn. teletype of in bepaalde gevallen te laten tekenen op een scherm dat door de computer kan worden afgelezen.

De leerstof-constructeur/programmeur kan diverse leerling-acties, diverse oplossingspogingen, plausibele fouten, verzoeken om hulp enz. anticiperen en hier adequate feedback en hints bij voorbereiden. Na aanbieding hiervan kan de leerling opnieuw actief oplossingspogingen ondernemen. Op deze wijze is een optimale 'responsive environment' te scheppen.

ad 7 Juist omdat rekening moet worden gehouden met de mogelijkheid dat de verschillende oplossingsmethoden niet dezelfde mate van wendbaarheid zullen bewerkstelligen moet niet alleen worden nagegaan hoe de leerprestaties op de post-test (de test die de getrainde problemen bevat) zich verhouden. Omdat het gaat om 'leren denken' is de prestatie op *transfer-tests* ook een belangrijke afhankelijke variabele. Daarnaast is het van belang dat niet een *tijdelijke* vaardigheid wordt aangeleerd maar een blijvende. Dit is een reden om na verloop van een aantal maanden zgn. *retentie-tests* af te nemen.

De transfer-tests moeten zo worden gekozen dat de inhoud meer en minder ver van de getrainde problemen ligt. Een handvat hiervoor (een soort heuristiek) vormt de 'Structure of Intellect' – conceptie van Guilford (1967).

Op de concrete inhoud van deze tests wordt in het vervolg van dit artikel ingegaan.

ad 8 De gekozen leerling-kenmerken (aptitudes) zijn:

- negatieve faalangst
- veldafhankelijkheid/veldonafhankelijkheid
- specifieke intelligentie factoren

De negatieve faalangst werd als predictor gekozen op grond van onderzoek van Hermans (1971) waarin geconstateerd werd dat negatief faalangstige leerlingen behoefte hebben aan sterk geprestructureerde leerstof. Deze leerlingen worden niet graag het bos ingestuurd met de opdracht zelf hun weg te bepalen. Omdat in de heuristische programma's de leerlingen in vrij hoge mate zelf hun weg moeten zoeken door de problemen (veel meer dan in de sterk gestructureerde algoritmische versie) zou interactie van deze persoonlijkheidstrek met de onderwijsprocedures (treatments) kunnen optreden.

Voor de motivering om de veldafhankelijkheid/veldonafhankelijkheid als aptitude-factor op te nemen, kan worden verwezen naar De Leeuw (1973).

Nadere beschrijving van de twee denkstofgebieden

1. Syllogismen-redeneringen

Iets ruimer gesteld kan dit ook omschreven worden als 'beoordelen van redeneringen op logische geldigheid'. Het gaat om het beoordelen of een conclusie logisch volgt uit de genoemde argumenten. Dit wordt getraind in de vorm van syllogismen sec. Deze vormen een abstractie van de logische redenering en argumentering zoals die voorkomt in een in tekst ingeklede vorm.

Syllogismen bestaan uit twee premissen en een conclusie. Beoordeeld moet worden of de conclusie geldig is, d.w.z. of deze logisch volgt uit de beide premissen. Premissen en conclusie kunnen op twee dimensies worden ingedeeld: bevestigend-ontkenkend en algemeen bijzonder. Binnen de klassebreedte-1-leerprogramma's worden syllogismen met uitsluitend algemene premissen aangeleerd. Deze kunnen zowel bevestigend (alle ... zijn ...) als ontkenkend (geen ... zijn ...) zijn. Binnen de klassebreedte-2-leerprogramma's komen naast algemene premissen ook bijzondere premissen voor (sommige ... zijn ... of sommige ... zijn geen ...). De problemen binnen elke klassebreedte worden volgens beide soorten oplossingsmethoden (algoritmisch en heuristisch) getraind.

In beide soorten oplossingsmethoden wordt van verzamelingenleer gebruik gemaakt. In de algoritmische versie wordt met Venn-diagrammen gewerkt; in de heuristische met Euler-cirkels. De algoritmische methode werkt met de begrippen lege en niet-lege verzameling, is het meest geformaliseerd en biedt oplossingsgarantie.

2. Getalrijen extrapoleren

In de klassebreedte-1-leerprogramma's worden de volgende typen getalrijen aangeleerd:

- getalrijen met vaste term/factor in de eerste afgeleide rij
- getalrijen met vaste term/factor in de tweede afgeleide rij (zgn. doorlopende regelmaat); d.w.z. getalrijen waarin de eerste afgeleide rij van type a is.

In de klassebreedte-2-leerprogramma's komen voor:

- + b. getalrijen van klassebreedte 1
- c. getalrijen met vaste term/factor in oneven/even eerste afgeleide rij (zgn. alternerende of om-en-om regelmaat)

Voorbeelden van de drie hoofdtypen a, b en c zijn:

Type A (vaste term/factor)

23	27	31	35	39	—	+4
128	64	32	16	8	—	:2

Type B (doorlopende regelmaat)

8	11	16	23	32	—	+3, +5, +7, +9
105	57	33	21	15	—	-48, -24, -12, -6
1	2	6	24	120	—	×2, ×3, ×4, ×5

Type C (alternerende regelmaat)

3	7	13	17	23	—	+4, +6
7	14	27	54	67	—	×2, +13
4	1	8	2	16	—	:4, ×8

Daarin is ook na te gaan dat de drie hoofdtypen weer bestaan uit een aantal subtypen. Zo zijn de getalrijen met doorlopende regelmaat nader onder te verdelen in getalrijen met doorlopende regelmaat in de rij van de verschillen (eerste onderrij) en getalrijen met doorlopende regelmaat in de rij van de factoren/delers (tweede onderrij). Binnen elk subtype is een nadere onderverdeling mogelijk in weer acht subtypen.

Vanwege de beschikbare ruimte zullen we ons beperken tot een nadere beschrijving van de getalrijenleerprogramma's. De syllogismenleerprogramma's komen dus niet aan de orde.

Voordat op de vormgeving en het werken met de leerprogramma's wordt ingegaan dient voor een goed begrip een korte beschrijving te worden gegeven van de gebruikte apparatuur.

Gebruikte apparatuur

Deze bestaat uit een DEC PDP-8 computer met verschillende typen terminals.

- TM 1024: een random-access projector waarin leer- en feedback-teksten op een 20 × 20 cm scherm worden geprojecteerd. Randomaccess betekent dat vanaf elke plaats op de film naar elke andere plaats kan worden gesprongen; dit onder computer-besturing.
- Beeldbuis: hierop kan door de leerling en door de computer worden geschreven en getekend.
- Teletype: een typemachine waarop zowel door de leerling als door de computer kan worden getypt. De leerling kan zijn antwoorden op de teletype intypen; ook kan zo om hulp of om beoordeling van een gegeven antwoord worden gevraagd. De computer zorgt voor sturing van deze terminals (op grond van een software programma). Ook wordt de registratie van de leergeschiedenis door de computer verzorgd.

Aanleren van algoritmische oplossingsmethoden

Over de wijze waarop het algoritmisch proces tot stand moet worden gebracht is Landa, de Russische specialist op het gebied van algoritmisering van het onderwijs, nogal vaag. Terecht stellen Van Parreren en Carpay (1972) dat de door Landa gekozen aanpak eclectisch en weinig consequent is. Wel komt in Landa's hoofdwerk (1969) een aantal stellingnamen naar voren. De belangrijkste daarvan zijn de volgende.

- Het is niet verstandig de algoritmen in kant en klare vorm aan te bieden. De leerling moet het algoritme òf zelf *afleiden* òf althans *begrijpen* waarom het algoritme deze vorm heeft.
- Het zich eigen maken van het algoritme (het leren beheersen) moet plaats vinden middels het *zelfstandig* oplossen van opgaven; het is onjuist om de leerling lange voorschriften in te prenten.
- Algoritmen mogen pas dan worden aangeleerd als de daarin voorkomende operaties *elementair* zijn voor de betreffende leerlingen; eenzelfde operatie kan voor de ene leerling elementair en voor de andere complex zijn. De complexe operaties dienen middels individueel onderwijs te worden 'omgeschoold' tot elementaire.
- Niet alle algoritmische processen hoeven middels, het *geven van een algoritme* te worden aangeleerd. Voor de identificatie-algoritmen geldt dat de logische structuur der kenmerken gekend moet worden en dat de leerling zelf uit de beschikbare verzameling van daarij passende algoritmen het meest rationele, meest efficiënte moet kiezen.
- Alle operaties binnen het algoritme moeten *actief* worden uitgevoerd; er moeten zoveel mogelijk gematerialiseerde handelingen voorkomen die dan later geïnterioriseerd worden. Hierin sluit Landa zich aan bij Galperin.

Landa heeft summier beschreven hoe hij m.b.v. geprogrammeerde instructie (de onderwijs-machine Repetitor I) algoritmische processen op grammatika-gebied heeft aangeleerd. Bij deze werkwijze laat zich een aantal vraagtekens plaatsen. Zo is het onduidelijk of foutmeldingen betrekking hebben op de door de leerling gemaakte *keuze* van de toetsing of op het door hem gegeven *antwoord* op de gekozen toetsing; het meest waarschijnlijk op grond van de beschrijving die Landa geeft, is dat de feedback op beide tegelijkertijd betrekking heeft. Dit zou voor het aanleren van een algoritmisch proces een dubieuze werkwijze zijn, gezien de weinig specifieke aard die de feedback dan heeft. Eveneens is het onduidelijk hoe het apparaat reageert als een irrelevante

toetsing wordt gekozen en beantwoord. Welke feedback volgt dan en wat is de betekenis daarvan voor de leerling?

Een andere onduidelijkheid: De leerling kiest uit operaties en toetsingen die beschreven staan op het apparaat. Hoe heeft geleidelijke uitwissing ('vanishing') van deze verbale steun plaats, zodat de leerling het algoritmisch proces tenslotte zelfstandig, zonder hulp en verbale aanwijzing voltrekken kan?⁴

Een ander punt van kritiek op de werkwijze van Landa staat vermeld in Van Parreren en Carpay (1972) hoofdstuk 5. Daarin staat het onderzoek van Kuljutkin en Suchobskaja beschreven. In het door hen getrainde algoritme bleek dat geen enkele proefpersoon werkte met een getrapte beslissingsprocedure, zoals die in de logische boom wordt weergegeven; er ontstaat dus geen algoritmisch proces. De ppn blijken simultaan en niet successief te beslissen. Deze kritiek stemt overeen met de kritiek die Sëchter (Van Parreren en Carpay, 1972) heeft uitgeoefend op het werk van Landa. Hierbij dient te worden aangekend dat de kritiek van Kuljutkin en Suchobskaja en vermoedelijk ook die van Sëchter alleen steek houdt als het gaat om algoritmen van een simpele structuur, bestaande uit een klein aantal stappen. Een voorbeeld om aan te tonen dat deze kritiek bij meer ingewikkelde algoritmen niet kan opgaan is te vinden in het algoritme voor de syllogismen-redeneringen. Dit algoritme bestaat uit 13 operaties en 14 toetsingen, waarbij een deel van de operaties dan nog bestaat uit subalgoritmen (een afbeelding kan wegens plaatsruimte niet worden getoond). Een algoritme van een dergelijke graad van ingewikkeldheid kan onmogelijk simultaan tot de juiste identificatie leiden. Blijkens een slotopmerking in het betreffende hoofdstuk is ook Kuljutkin van mening dat in zo'n geval boomschema's een nuttige functie kunnen hebben. Wel is het zaak om uit deze kritieken de les te trekken dat het leerprogramma zodanig moet zijn samengesteld, een zodanig 'Lehralgoritme' moet bevatten, dat de leerling tot het successief volgen van de stappen uit het algoritme gedwongen wordt.

Deze overwegingen hebben ons geleid tot de keuze van een expositorische onderwijsprocedure waarin sterk sturend te werk wordt gegaan.

Hoe de bovengenoemde uitgangspunten en punten van kritiek zijn verwerkt in het algoritmisch leerprogramma getalrijen zal hierna worden beschreven.

Het algoritme 'getalrijen aanvullen'

Het vereenvoudigd algoritme voor het opsporen en extrapoleren van de regelmaat in de vermelde typen

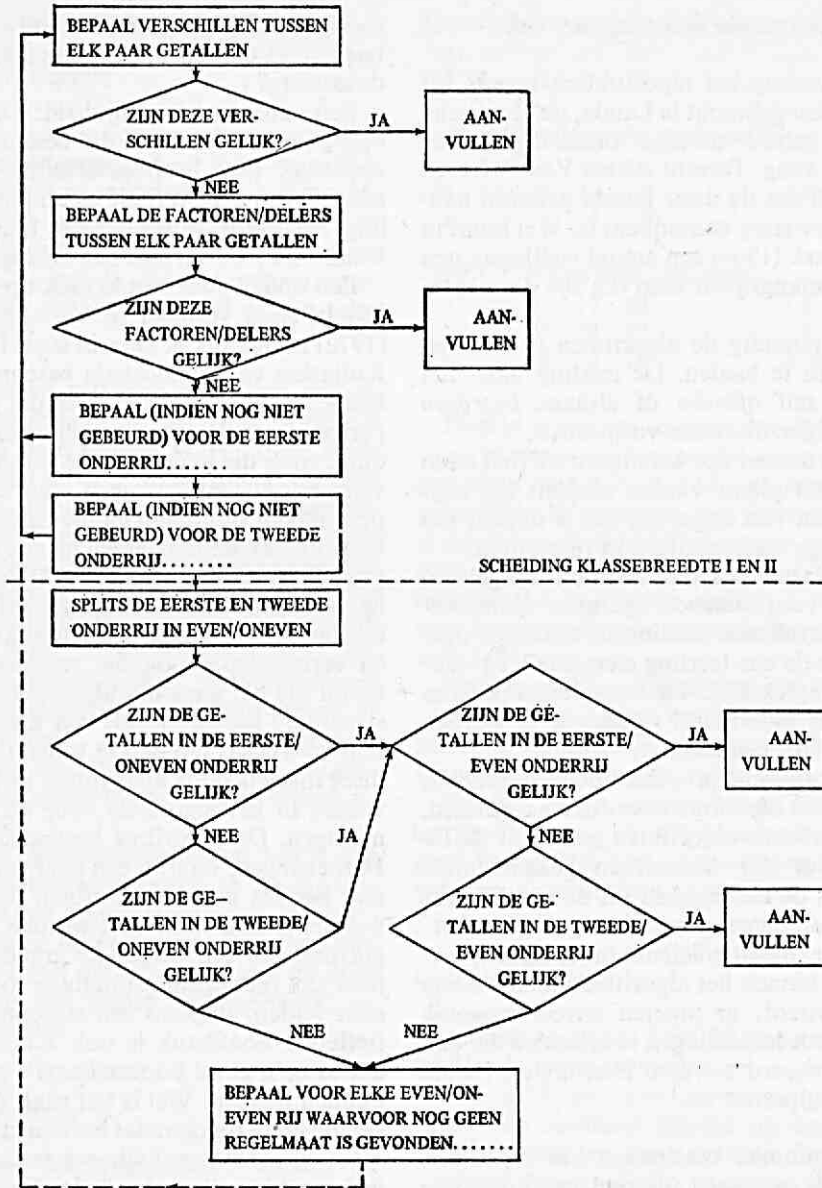


Fig. 1 vereenvoudigd algoritme getalrijen

getalrijen is weergegeven in Figuur 1. De horizontale stippellijn geeft de scheiding weer tussen de klassebreedten 1 en 2. Het gedeelte boven de stippellijn vormt dus het algoritme voor klassebreedte 1 (type a + b); het totaal vormt dus het algoritme voor klassebreedte 2 (type a t/m c).

De operaties en toetsingen zijn resp. door rechthoeken en ruiten weergegeven. Vanaf de onderste rechthoek loopt een stippellijn die verwijst naar de bovenste rechthoek. Deze lijn is gestippeld omdat

het soort regelmaat waarnaar dan gezocht wordt, voorkomt in een type rij dat buiten klassebreedte 2 valt.

Dit type is opgenomen als transfer-opgave (zie later). Het algoritme heet vereenvoudigd omdat bepaalde stappen elk op zich weer bestaan uit subalgoritmen die apart dienen te worden ingeoeffend. Dit geldt o.a. voor de operatie 'aanvullen'. Dit aanvullen is een transformatie-(sub)algoritme dat op zich weer bestaat uit een aantal stappen.

Voorbeeld: 3 4 7 16 43
 +1 +3 +9 +27 .. --

De getallen in de afgeleide rij worden hier steeds drie maal zo groot (doorlopende regelmaat). Dan moet eerst het getal op de stippen worden bepaald (middels het vermenigvuldigen van de juiste getallen met elkaar); dan moet het getal op de streep worden berekend (middels optellen).

Het algoritme is opgebouwd uit identificatie- en transformatiegedeelten. Het gaat in termen van Kuljutkin om het opsporen van de explicatieve relatie (identificatie); daarna volgt het implicatief proces (transformatie = extrapoleren).

Het algoritmisch leerprogramma 'getalrijen aanvullen'

Voorafgaand aan de eigenlijke constructie van het leerprogramma moet worden beslist volgens welke van de volgende twee hoofdvormen het algoritmische proces tot stand zal worden gebracht:

a. Per getalrij het algoritme volledig laten afwerken d.w.z. voor de betreffende getalrij alle operaties en toetsingen achtereenvolgens laten uitvoeren. Als dit – onder programma controle – voor alle stappen van de betreffende getalrij heeft plaatsgevonden, de volgende rij aanbieden en ook hiervoor het algoritme volledig laten afwerken enz. tot het criterium is gehaald.

b. Eerst de verschillende stappen *apart* inoefenen, dus steeds nadat de betreffende stap is uitgevoerd een nieuwe getalrij aanbieden en hiervoor dezelfde stap laten uitvoeren; dit voortzetten tot het criterium wordt gehaald. Pas als de stappen apart volledig worden beheerst wordt het algoritme *in zijn geheel* geoefend.

Er zijn tussenvormen van a en b mogelijk. Zo kan begonnen worden met het inoefenen van stap A, daarna kan AB worden ingeoefend; daarna ABC. Of de volgorde kan zijn, A, B, AB, C, ABC enz. Uitgaande van de door Landa genoemde wenselijkheid dat de operaties van het algoritme *elementair* zijn hebben wij gekozen voor b⁵.

Dit leidt tot de volgende programma-onderdelen:

1. INTRODUCTIE
2. Sub-algoritme BEREKENEN ONDERRIJEN
3. Sub-algoritme TOETSING
4. Sub-algoritme AANVULLEN
5. TOETSINGSKEUZE + TOETSING
6. TOETSINGSKEUZE + TOETSING + AANVULLEN

7. Volledig algoritme:

BEREKENEN ONDERRIJEN + TOETSINGSKEUZE +
TOETSING + AANVULLEN

8. NA-TEST

Het algoritme (schema) dat met de leerlingen wordt opgebouwd is van een wat andere vorm dan het in Figuur 1 geschetste algoritme.

Zo wordt in het schema aan de hand waarvan de leerlingen leren werken vragen gesteld als 'bevat de tweede onderrij een doorlopende regelmaat?'. De beantwoording van deze vraag vereist uitvoering van een aantal operaties en toetsingen uit Figuur 1.

Het feit dat dit schema grover is dan het in Figuur 1 afgebeelde maakt dat de betreffende stappen minder elementair zijn. Om deze stappen voor alle leerlingen elementair te maken worden deze apart ingeoefend. Anders gesteld: de betreffende stappen uit het schema zijn te beschouwen als sub-algoritmen of subroutines. Geheel in lijn met Landa (1969), hoofdstuk 10, zijn wij van mening dat dergelijke subalgoritmen niet via een visueel schema hoeven te worden ingeoefend, maar dat de logische structuur der kenmerken middels systematische feedback kan worden aangeleerd.

ad 1 INTRODUCTIE: Dit onderdeel wordt middels de TM 1024 aangeboden; de vragen worden via de teletype beantwoord. Deze introductie doet dienst als een soort 'advance organizer' (Ausubel, 1963). Een aantal later te gebruiken begrippen wordt toegelicht t.w. de begrippen regelmaat, eerste en tweede onderrij (resp. de rij van de verschillen en van de factoren/delers), verschillende soorten regelmaat; ook wordt gewezen op de noodzaak om, ten einde het soort regelmaat op te sporen, een aantal vragen in de juiste volgorde te beantwoorden. Dit wordt geoefend. Daarna wordt het algoritme geïntroduceerd (de term 'schema' wordt gebruikt tegenover de leerlingen). Het schema-gebruik wordt geoefend. De leerlingen beantwoorden voor een aantal getalrijen de vragen uit het schema met ja/nee. Er wordt hier niet tot een criterium gewerkt. Het schema wordt uitgebreid als blijkt dat het voor bepaalde getalrijen niet tot opsporing (identificatie) van de daarin voorkomende regelmaat leidt. De bedoeling van deze introductie is om het algoritme niet in kant en klare vorm aan te bieden maar om dit *samen met de leerlingen* op te bouwen. Bovendien moet de leerling vanaf de eerste leertekst die wordt aangeboden vragen beantwoorden met ja/nee of antwoorden intypen, zodat ze voortdurend *actief* bezig zijn.

De leerling ontvangt feedback over zijn antwoord-

den, d.w.z. dat door het onderwijsprogramma op goede antwoorden bevestigend wordt gereageerd en op fouten middels een corrigerende tekst wordt ingegaan.

ad 2 Sub-algoritme BEREKENEN ONDERRIJEN

De introductie wordt, voordat het schema wordt aangeboden, onderbroken om het berekenen van eerste en tweede onderrij te oefenen tot een criterium is gehaald. Daartoe verschijnen de getalrijen op het beeldscherm. De leerling moet nu beginnen om voor de betreffende getalrij het eerste getal van de eerste onderrij te berekenen. Er zijn hiervoor standaard-feedback-teksten geconstrueerd zoals 'het getal is goed maar het teken is fout. Druk op de VEEGUIT toets, dan verdwijnt je foute antwoord van het scherm en kan je het opnieuw proberen'. De leerling wordt niet door het leerprogramma losgelaten voordat voor de betreffende getalrij alle verschillen en alle mogelijke factoren/delers correct op het scherm staan. Het programma heeft een 'drill-and-practice'-achtig karakter en is bedoeld om deze stap in termen van Landa tot een elementaire operatie te maken. Pas als voor een bepaald aantal getalrijen (het criterium) *foutloos* de beide onderrijen zijn berekend, wordt naar de volgende fase van het leerprogramma overgegaan.

ad 3 Sub-algoritme TOETSINGEN

Direct na het tweede deel van de introductie worden de in het schema voorkomende toetsingen geoefend (voor klassebreedte 1 zijn dat er vier; voor klassebreedte 2 zijn dat zeven verschillende toetsingen). Het gaat dus om het beantwoorden van de ja/nee vragen uit het schema. Reeds eerder is gesteld dat binnen de soorten regelmaat waar in het schema naar gevraagd wordt een aantal subtypen kunnen worden onderscheiden. Zo zijn er 8 sub-typen doorlopende regelmaat in de eerste onderrij te onderscheiden.

Het begrip 'doorlopende regelmaat in de eerste onderrij' is dus een disjunctief begrip. De logische structuur der kenmerken voor dit begrip wordt hier niet aangeleerd door een visueel schema te gebruiken, maar door systematische feedback te verschaffen op vragen als: 'Heeft de eerste onderrij een doorlopende regelmaat?' of 'Is er een kruislingse om-en-om regelmaat aanwezig?'

De vragen en feedback-teksten worden op de TM-1024 projector aangeboden. Elk der 4 of 7 toetsen wordt geoefend tot een criterium. Als dit is gehaald dan worden de soorten toets-vragen door elkaar gesteld; ook weer tot een criterium is gehaald,

zodat voor *elke* leerling de betreffende algoritme-stap elementair is na afloop van dit onderdeel.

ad 4 Sub-algoritme AANVULLEN

In het schema komt herhaaldelijk de term AANVULLEN voor. Dit steeds uitgaande van een gevonden type regelmaat. Ook dit aanvullen is niet voor alle leerlingen een elementaire operatie. Het is een transformatie-(sub)algoritme dat apart dient te worden ingeoefend.

De te extrapoleren getalrijen verschijnen op het beeldscherm. Daarbij wordt het soort regelmaat dat in de rij voorkomt eronder geschreven bijv. 'om-en-om regelmaat in de eerste onderrij'. De leerling moet nu het getal boven de pijl intypen, d.i. het getal dat in één der beide onderrijen moet staan om de rij te kunnen aanvullen. De feedback - die op de TM 1024 verschijnt - sluit aan bij het gemaakte type fout. Er zijn 16 soorten teksten.

Een voorbeeld van zo'n tekst die wordt aangeboden als de leerling het getal -8 intypt bij de getalrij.

64	32	24	12	4	--
-32	-8	-12	-8		
:2	--	:2	:3		
				↑ □	

'Het getal dat je hebt getypt hoort wel bij de om-en-om regelmaat van de rij, maar dat getal *is niet aan de beurt*. Het staat één plaats links van de pijl namelijk ook al.

In een om-en-om regelmaat kan eenzelfde getal *niet twee keer direct achter elkaar* voorkomen.

Druk maar op de veeg-uit knop, dan verdwijnt je foute antwoord uit het hokje. Typ daarna het goede antwoord (denk om de beoordeel-knop en eventueel weer de veeg-uit-knop). Als je het echt niet meer weet dan mag je een ? typen; dan krijg je hulp. Maar probeer het liever zelf.'

In geval van een fout moet de leerling zijn antwoord middels een druk op de knop uitwissen en opnieuw proberen, net zo lang tot het goede getal boven de pijl staat. Is de leerling niet in staat het goede antwoord te vinden dan kan hij middels het indrukken van een ?-knop om hulp vragen.

Hij krijgt dan de plaats waar het getal boven de pijl moet staan (in eerste of in tweede onderrij) gegeven. Bij opnieuw om hulp vragen wordt het getal boven de pijl door de computer op het scherm geschreven. De criterium-teller komt dan, evenals bij een fout, op nul te staan.

Als - met of zonder hulp - het getal boven de pijl op het scherm staat dan moet de leerling het getal

op de streep intypen. Ook hier wordt net zo lang doorgedaan tot het goede antwoord is getypt. De feedback luidt hier alleen goed/fout.

ad 5 TOETSINGSKEUZE + TOETSING

De bedoeling van dit programma-onderdeel is te komen tot identificatie van het type regelmaat.

Dit onderdeel is volgens de volgende uitgangspunten samengesteld:

- De betreffende identificatie dient tot stand te worden gebracht door een algoritmisch proces (zoals door het algoritme (schema) beschreven) te leren voltrekken.
- De leerling dient het algoritmisch proces tenslotte zonder schema-hulp te voltrekken.
- De keuze van de juiste toetsing en het beantwoorden van de toetsingsvragen dient in het leerprogramma wel te worden onderscheiden maar niet te worden gescheiden.

Deze punten zijn als volgt in de programma-structuur verwerkt:

De leerling kiest aanvankelijk de aan de orde zijnde toetsingsvraag met behulp van het schema. Hij doet dat door van de vier of zeven (resp. voor klassebreedte 1 en 2) omschrijvingen van de toetsingen (steeds in een andere random volgorde gepresenteerd) aan te geven welke aan de beurt is; dit door het getal te typen dat voor de door hem gekozen toetsing staat. Hij krijgt over zijn keuze feedback. Deze feedback is zodanig samengesteld, dat de leerling leert geen stappen in het schema over te slaan. Als de leerling voorbarig - d.w.z. voordat de betreffende toetsing volgens het schema aan de orde is - de toetsing kiest die behoort bij de regelmaat van de betreffende rij, dan wordt hij terechtgewezen.

Dit om een getrapte beslissingsprocedure (algoritmisch proces) tot stand te brengen en zo de ook door Kuljutkin en Suchobskaja genoemde mogelijkheid van streven naar simultane identificatie (met alle mogelijke fouten van dien) uit te sluiten.

De bedoeling is uiteraard dat de leerling tenslotte de toetskeuzes in de juiste volgorde foutloos kan uitvoeren zonder schema, of beter gesteld: met een geïnterioriseerd schema. Het gebruik van het visuele schema dient daartoe geleidelijk te worden weggenomen ('gevanished'). Eén van de punten van kritiek op de werkwijze van Landa was, dat niet duidelijk is hoe hij tot 'vanishing' van het algoritmische schema komt. Hoe deze 'vanishing' in ons leerprogramma is gerealiseerd moge blijken uit het volgende.

Als derde uitgangspunt geldt, dat er duidelijk onderscheid dient te worden gemaakt tussen het leren kiezen van de goede toets en het geven van het goede antwoord daarop. In programma-onderdeel 3

is het toetsen apart geoefend. In het nu aan de orde zijnde onderdeel worden toetskeuze en toetsen tenslotte gecombineerd geoefend.

Bovenstaande uitgangspunten hebben geresulteerd in een indeling van dit programma-onderdeel in drie niveaus:

- a. toetsingskeuze met schema zonder toetsing
- b. toetsingskeuze zonder schema zonder toetsing
- c. toetsingskeuze zonder schema met toetsing

Op elk niveau wordt gewerkt met een criterium. Pas als dit gehaald is, wordt naar het volgende niveau overgegaan.

Als gewerkt wordt binnen niveau b of c (zonder schema) en de leerling kiest de verkeerde toetsingsstap, dan wordt bij de feedback het schema weer getoond, zodat de leerling de juiste stap met behulp van het schema kan kiezen. De criterium-teller gaat dan echter op nul. Pas als alle stappen binnen een getalrij foutloos zijn uitgevoerd wordt de criterium-teller met 1 verhoogd.

Binnen niveau a en b worden de toetsingsvragen door het programma beantwoord. Binnen niveau c moet de leerling deze vragen beantwoorden.

De getalrijen aan de hand waarvan het algoritmisch proces dient te worden voltrokken, worden (incl. de onderrijen) op het beeldscherm aangeboden.

ad 6 + 7 (BEREKENEN ONDERRIJEN) + TOETSINGSKEUZE + TOETSING + AANVULLEN

In onderdeel 6 en 7 worden gaandeweg meer delen van het algoritme tegelijk geoefend. In onderdeel 7 dienen de leerlingen tenslotte het volledige algoritmische proces bij steeds andere getalrijen te voltrekken. Dit weer tot aan een criterium wordt voldaan.

ad 8 NA-TEST

Om er zeker van te zijn dat de leerlingen ook zonder de betreffende vragen en opdrachten van het leerprogramma het algoritmisch proces voltrekken, wordt in het laatste onderdeel een aantal getalrijen ter oplossing aangeboden, zonder dat er enige sturing van het gedrag optreedt. De leerlingen typen het getal waarmee de rijen dienen te worden aangevuld in (zonder enige tussenstap aan de computer mede te delen). Als er teveel fouten worden gemaakt, dan wordt, op grond van de veronderstelling dat de leerling het algoritmisch proces niet (steeds) volledig heeft voltrokken, terugverwezen naar onderdeel 7 van het leerprogramma. Is daar het criterium dan opnieuw gehaald, dan wordt weer overgegaan naar onderdeel 8. Hier herhaalt de procedure zich (iteratieve procedure).

Aanleren van heuristische oplossingsmethoden

Zoals reeds eerder vermeld kan men binnen de heuristische oplossingsmethoden graden van algemeenheid onderscheiden. Men kan een methode heuristisch noemen als daarin zeer vage aanwijzingen worden gehanteerd in de zin van: 'probeer het probleem eens van een andere kant te benaderen' of 'kijk eens wat je met de gegevens kunt doen' of 'probeer het doel eens anders te formuleren'. Daarentegen zijn methoden waarin een veel grotere mate van sturing aan het denkproces wordt gegeven ook heuristische methoden.

Als men nu het effect van een algoritmische methode wil vergelijken met een heuristische methode, dan zal men op het continuüm 'heuristische methoden' min of meer arbitrair een graad van specificiteit van die heuristische methode moeten kiezen.⁶

De heuristische methode waarvoor wij gekozen hebben bevat een vrij grote mate van sturing van de denkprocessen; daarin worden tamelijk specifieke aanwijzingen en hints gehanteerd.

Omdat het de bedoeling is dat de leerling zichzelf de genoemde aanwijzingen en hints leert verschaffen en hiermee leert werken, kortom de heuristische oplossingsmethode leert hanteren, moet het gebruik maken van dit soort hints in het onderwijsprogramma worden geoefend. Het onderwijsprogramma moet bovendien stimuleren tot het blijven zoeken naar een oplossing via andere wegen. Dit teneinde het in het onderwijsprogramma te oefenen gedrag het doelgedrag zo dicht mogelijk te laten benaderen.

Het onderwijsprogramma zal daarom het karakter hebben van geleide ontdekking ('guided discovery') en zal daarom zelf heuristisch van aard zijn.

Het heuristisch leerprogramma 'getalrijen aanvullen'

Belangrijk voor het correct aanvullen van getalrijen is dat leerlingen in staat zijn om op grond van de verbanden die ze tussen bepaalde getallen zien tot hypothesen te komen over het soort regelmaat dat in de rij aanwezig is. Anders gezegd: belangrijk is dat ze op grond van relaties die ze zien komen tot de *explicatieve* relaties (Kuljutkin, 1970).

Belangrijk is ook, dat de hypothesen die ze genereren *getoetst* worden. Zo blijken leerlingen veel fouten te maken doordat ze hypothesen opbouwen op grond van het eerste of laatste getallenpaar van de rij en deze hypothese niet toetsen aan de overige delen van de rij.⁷

Niet alleen is het belangrijk dat hypothesen systematisch en consequent worden getoetst; zeker zo

belangrijk voor het bereiken van de oplossing is, dat als de betreffende hypothese dient te worden verworpen, een *nieuwe* hypothese wordt bedacht. Dit kunnen genereren van steeds nieuwe hypothesen is van groot belang. Het is een vorm van divergent denken. Een heuristische aanpak dient gericht te zijn op de bevordering van dit divergente denken en op het systematisch toetsen welke van de mogelijke relaties de explicatieve relaties vormen (convergent denken). Het leerprogramma dient erop gericht te zijn dit gedrag van de leerling tot stand te brengen. Dit natuurlijk met de bedoeling dat de leerling dit gedrag tenslotte ook vertoont buiten de steun van het leerprogramma om. De leerling moet de sturing leren vervangen door zelfstandige organisering van zijn gedrag, moet zichzelf vragen leren stellen en zelf ideeën leren genereren.

Dit is in het heuristische leerprogramma als volgt nagestreefd:

De leerling krijgt niet, zoals in het algoritmische programma, een volledige getalrij, maar de eerste drie getallen van een rij. De problemen worden via de TM 1024 als volgt gepresenteerd:

'Ik heb een getallenrij in mijn gedachte met een bepaalde regelmaat erin. Jij krijgt van mij de eerste drie getallen van mijn rij. Nu moet je door verstandig te raden erachter zien te komen welk getal in mijn rij op de plaats van de streep staat'.

Een voorbeeld: 4 6 12 —

De eerste drie getallen kunnen meestal op *meerdere* manieren geëxtrapoleerd worden. Als de leerling zijn antwoord intypt, volgt de feedback. Die feedback kan zijn: 'Je antwoord is goed, dit getal staat inderdaad in mijn rij'. De leerling krijgt dan vervolgens de opdracht om het volgende getal in de rij te vinden. Vaker zal — als een verstandig antwoord gegeven is — de feedback zijn: 'je hebt een verstandig antwoord gegeven, maar in mijn rij staat toch een ander getal, probeer het nog eens'. Als de leerling een onverstandig antwoord geeft (d.w.z. dat er geen regelmaat in zijn rij zit), dan is de feedback negatief. Als de leerling geen andere verstandige oplossingen meer ziet dan mag hij om hulp vragen door de 'help'-toets in te drukken. Er volgt dan een hint in de juiste richting.

Voordeel van deze benadering is, dat de leerling gedwongen wordt om, zodra hij hoort dat het door hem getypte getal niet het juiste getal is, zijn hypothese te laten vallen en een nieuwe hypothese te bedenken en te toetsen. Mocht deze weer onjuist

blijken dan herhaalt het proces zich. Het gebruik van de 'help'-toets heeft het voordeel dat de leerling uit zijn mogelijke fixatie aan een bepaalde hypothese (hij denkt bijv. uitsluitend aan een om-en-om regelmaat en komt niet op het idee van een doorlopende regelmaat) wordt losgemaakt, zonder dat hij het goede antwoord direct gegeven krijgt. Het zal duidelijk zijn, dat hiermee een grotere flexibiliteit, een grotere wendbaarheid van het denken van de leerling beoogd wordt. Dit kan *transfer-bevorderend* werken als onbekende typen getalrijen moeten worden aangevuld.

Deze aanpak is vooraf op een drietal scholen voor basisonderwijs van verschillend sociaal-economisch niveau mondeling uitgetest (proefleider met individuele leerling). Uit de resultaten en reacties van de leerlingen bleek dat dit een vruchtbare werkwijze is. Het C.A.I.-programma moet worden gezien als een simulatie van deze leraar-leerling interactie.

Hieronder willen we iets nader ingaan op de inhoud en structuur van het heuristische leerprogramma. We willen dat doen aan de hand van het getalrijvoorbeeld hierboven.

De constructeur van het onderwijsprogramma moet anticiperen op verschillende oplossingspogingen van de leerling en daarbij adequate feedback gereedmaken.

De te verwachten antwoorden zijn te verdelen in vier soorten:

1. plausible (verstandige) antwoorden
hier: 22 (+2, +6, +10)
14 (+2, +6, of +2, ×2)
2. 'begrijpelijke' foute antwoorden
hier: 24 (12 × 2)
18 (12 + 6)
3. Het 'goede' antwoord
hier: 30 (+2, +6, +18)
4. 'onbegrijpelijke' foute antwoorden
hier: alle hierboven niet genoemde getallen

Hierbij moet worden opgemerkt dat het goede antwoord (hier: 30) arbitrair als het goede antwoord is gekozen. Daarom zijn de onder 1. genoemde antwoorden niet minder goed; het worden dan ook verstandige antwoorden genoemd.

Omdat als uitgangspunt van het heuristisch leerprogramma geldt dat de leerling de relaties zoveel mogelijk zelf moet ontdekken, moet, als de leerling vastloopt, de mogelijkheid bestaan hem om hulp te laten vragen middels het indrukken van een 'help'-knop. Die hulp mag niet inhouden dat het goede antwoord zonder meer *gegeven* wordt. Dit geven

van het goede antwoord mag pas in laatste instantie plaats hebben en wel als na een cumulatieve 'prompting' nog steeds geen oplossing is gevolgd. Tussen het geven van de eerste 'prompt' (hint) en het tenslotte geven van het goede antwoord liggen hints die steeds sterker in de richting van de oplossing leiden.

Als de leerling om hulp vraagt wordt zoveel mogelijk aangesloten bij al eerder gegeven plausible antwoorden. Als al meerdere plausible antwoorden gegeven zijn dan kan het ene meer wijzen in de richting van het soort regelmaat dat de computer in gedachten heeft dan het andere. In het voorbeeld hierboven heeft de computer een rij met een doorlopende regelmaat van een bepaald soort (+2, +6, +18) in gedachten. Van de twee mogelijke plausible antwoorden (22 en 14) is de eerste ook gebaseerd op een doorlopende regelmaat (+2, +6, +10), maar dan met een vaste term i.p.v. een vaste factor in de tweede afgeleide rij. De leerling zit, als hij al ooit 22 heeft geantwoord daarmee dicht in de buurt van de gezochte regelmaat, veel dichter dan het soort regelmaat (om-en-om) waar het getal 14 op gebaseerd is. Het is didactisch waardevol om als nu hulp geboden wordt, aan te sluiten bij die oplossingspoging die het meest in de richting van de gezochte regelmaat leidt. Om deze reden is in het hulp-gedeelte een *prioriteiten*-volgorde ingebouwd. In het voorbeeld hierboven wordt zodra de leerling om hulp vraagt eerst gekeken of hij al ooit 22 geantwoord heeft; als dat het geval is dan wordt de hulp geboden die daarbij aansluit, d.w.z. de leerling wordt gewezen op het bestaan van een ander soort *doorlopende* regelmaat in de eerste onderrij. Als nog nooit 22 is geantwoord dan wordt nagegaan of al ooit 14 is geantwoord; als dat zo is dan wordt hierbij aangesloten, d.w.z. de leerling wordt gewezen op het bestaan van andere soorten regelmatigheden dan om-en-om regelmatigheden. Als geen der plausible antwoorden ooit gegeven is, dan volgt een meer algemene hint.⁸ In alle gevallen van 'eerste hulp' worden de onderrijen gegeven.

Na de hulp kan de leerling nieuwe oplossingspogingen doen. Hij heeft ook de vrijheid om direct na de eerste geboden hulp of na één of meer oplossingspogingen opnieuw om hulp te vragen. Tenzij hij dan intussen een antwoord heeft gegeven dat hoger op de prioriteitenlijst staat dan het antwoord op grond waarvan de vorige keer hulp werd verstrekt, wordt hem een *sub(hulp)-probleem* aangeboden.

Dit hulp-probleem behelst de opdracht om het getal in één der onderrijen te geven.

In het voorbeeld: 4 6 12 --
 +2 +6 ..
 -- ×2

Ook binnen dit sub-probleem bestaat de mogelijkheid om hulp te vragen. In dat geval geeft de computer het antwoord.

Zodra het sub-probleem is opgelost wordt teruggegaan naar het hoofdprobleem en moet alsnog het getal op de streep worden gevonden.

Hierboven is de procedure geschetst die wordt gevolgd om de leerling te brengen tot oplossing van de betreffende getalrij. Als nu hierna nieuwe getalrijen ter oplossing zouden worden aangeboden, zouden daarbij andere beslissingsstructuren kunnen worden gehanteerd. Dan zou echter voor elk nieuw probleem dat ter oplossing wordt aangeboden een nieuw sturings(software)-programma en geheel nieuwe feedback-teksten (courseware) moeten worden samengesteld. Dit zou uiterst arbeidsintensief zijn. Bovendien – en belangrijker – zou de heuristische aanpak dan sterk heterogeen van aard en daardoor slecht geoperationaliseerd en gestandaardiseerd zijn.

Om deze redenen is gezocht naar een standaard-beslissingsstructuur en een aantal standaard feedback-skelet-teksten. Die standaardbeslissingsstructuur wordt ook wel 'teaching logic' of 'Lehralgorithme' genoemd. Het bevat de beslissingsregels op grond waarvan de leerling afhankelijk van zijn reacties – niet alleen de direct voorafgaande, maar ook zijn leergeschiedenis – door het leerprogramma wordt gestuurd.

Het gebruik van een dergelijke 'teaching logic' mag niet ten koste gaan van de didactische kwaliteit van het programma. Dit zou het geval zijn als de feedback die de leerling krijgt minder specifiek wordt, minder direct aansluit bij het gegeven antwoord, de gemaakte denkfout. Dit zou ook het geval zijn als de hulp die de leerling krijgt minder éénduidig wordt voor de leerling of minder aansluit op al eerder ondernomen oplossingspogingen. Dit is voor het heuristische getalrijenprogramma ondergaan door te werken met een soort skelet-teksten waarin de open ruimten per getalrij nader worden gespecificeerd.

De betreffende 'teaching-logic' kan hier wegens plaatsgebrek niet worden getoond. Daarvoor zij verwezen naar De Leeuw (1975a) en De Leeuw (1975b).

De transfer-tests

Uitgaande van de conceptie dat het aanleren van

oplossingsmethoden een middel is om te leren denken, is het wenselijk na te gaan of de geleerde vaardigheden ook overdracht, transfer vertonen naar verwante gebieden. De transfer-opgaven dienen in een bepaalde mate gerelateerd te zijn aan de geoefende problemen. De keuze van de transfer-tests vormt een probleem op zichzelf of zoals Elshout (1973) het stelt: 'De wijze waarop men dit keuzeprobleem oplost is in transfer-onderzoek van cruciaal belang: door de keuze van de situaties waarin men transfer-effecten meet wordt in hoge mate bepaald of men transfer-effecten vindt en tevens wordt erdoor bepaald hoe men deze effecten (of het ontbreken ervan) moet interpreteren'. Ideaal is de transfer-tests systematisch te kiezen, d.w.z. volgens een theoretisch model. Uitgaande van de idee dat de beide geoefende probleem-soorten ('syllogismen redeneren' en 'getalrijen extrapoleren') elk een intelligentie-aspect vertegenwoordigen, lijkt het intelligentie-model van Guilford (1967) een systematisch middel om tot keuze van transfer-tests te komen. Binnen dit 'Structure of Intellect' model is het syllogismen-redeneren te beschrijven als 'evaluatie van semantische relaties'; het getalrijen extrapoleren als 'convergente produktie van symbolische relaties'. Het zou voor de hand liggen om als transfer-tests die intelligentie-aspecten te kiezen die in de ruimtelijke nabijheid liggen van het kubusje dat de inhoud van het leerprogramma beschrijft. Men zou de betreffende transfer-tests dan systematisch meer en minder ver van de leerprogramma-inhoud kunnen kiezen.

Het systeem van Guilford is echter niet exact genoeg om op deze systematische wijze te werk te mogen gaan. Wel kan het model als heuristisch worden gebruikt bij de keuze. Of zoals Cronbach en Snow (1969) het stellen: '... two cells that are similar with respect to two or three facets are not hypothesized to have any closer relationship than cells having no similarity' en 'If we understand correctly, the famous 'box' is no more than a heuristic that suggests cells where factors should be found'.

De te kiezen transfer-opgaven dienen verwant te zijn met de geoefende problemen. Wij hebben een onderscheid gemaakt in verwantschap in *type operatie* en in *inhoud*. Parallel hiermee loopt dan de indeling in *horizontale* en *vertikale* transfer. Daarnaast wordt *formele* transfer onderscheiden.

Horizontale transfer-opgaven zijn dan opgaven die hetzelfde soort operaties vereisen maar inhoudelijk anders zijn dan de geoefende. Vertikale transfer-opgaven hebben dezelfde inhoud maar slechts een deel van de vereiste operaties zijn geoefend. Bij deze verticale transfer-opgaven wordt dus wel voortgebouwd op aangeleerde operaties maar deze zijn niet

toereikend. Zie bijv. het algoritme voor de getalrijen (Figuur 1); de stippellijn geeft dit voortbouwen op geleerde operaties weer.

Formele transfer-opgaven vereisen een geheel ander type operaties; de inhoud kan verwant zijn aan die van de aangeleerde opgaven. De formele transfer-opgaven kunnen meer of minder sterk met de getrainde opgaven verwant zijn. Een zeer vage verwantschap is aanwezig als het gemeenschappelijke relevante kenmerk van getrainde- en transfer-opgaven iets is in de zin van 'letten op alle elementen in de probleemsituatie', zoals dat door Andrae (Selz, 1935) werd beschreven. Vage verwantschap in deze zin hoeft niet in te houden dat geen transfer zal optreden, naar opgaven die alleen dit kenmerk met de geoefende opgaven gemeen hebben, zoals ook door Andrae is aangetoond. Sterkere verwantschap lijkt aanwezig als het gaat om opgaven die inhoudelijk gelijksoortig zijn en waarvoor de geleerde operaties niet relevant zijn.

Op grond van deze criteria kunnen voor de getalrijen de volgende transfer-tests worden onderscheiden:

Horizontale transfer-tests:

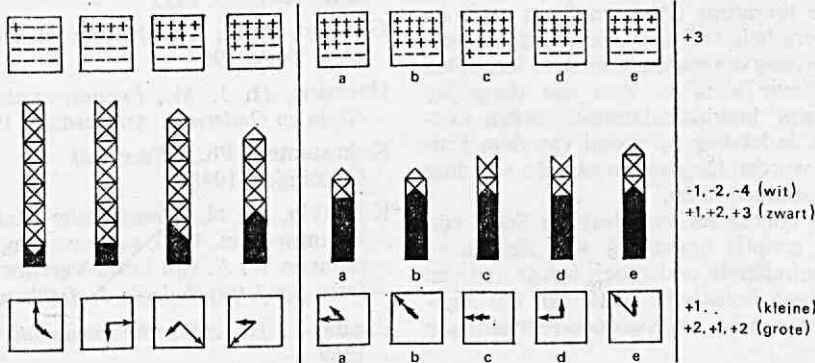
Interpoleren en terugredeneren binnen de geoefende typen getalrijen.

Voorbeelden:

3	19	—	51	67	83		+16
—	2	6	24	120	720		×2, ×3, ×4, ×5, ×6
2	8	4	—	12	48		×4, -4, ×4, -4

Figurenreeksen met hetzelfde soort regelmaat als de geoefende typen getalrijen.

Voorbeelden:



Vertikale transfer-tests:

Voor klassebreedte 1:

Getalrijen met om-en-om regelmaat

Voor klassebreedte 1 en 2:

getalrijen met doorlopende regelmaat in oneven/even onderrij

getalrijen met doorlopende regelmaat in de tweede afgeleide rij

Voorbeelden resp.:

72	56	36	28	18	14		-16, -20, -8, -10, -4	:2
3	8	17	33	59	98		+5, +9, +16, +26, +39	+3
							+4, +7, +10, +13	

Formele transfer-tests:

Getalrijen waarvoor de onderrijen niet relevant zijn

Voorbeelden:

35	25	30	5	25	1		-5, :5
2	3	5	8	13			+

Subtest Tekens Invullen van de GALO

Subtest Getallen van de GALO

Subtest Verbale Analogieën van de GALO

Slotopmerkingen

In dit artikel is de evaluatie van de leerprogramma's en het differentieel effect van de verschillende versies op de criteriumtests niet aan de orde gekomen. De aandacht is gericht geweest op vormgeving van de onderwijsprogramma's en in mindere mate op het gebruik van computersturing van deze onderwijsprogramma's.

Het uitproberen van de programma-versies heeft op een tiental leerlingen plaats gehad. We willen hier volstaan met te vermelden dat deze leerprogramma's naar behoren functioneren en geen verbetering behoeven. De aanwijzingen en instructies blijken duidelijk te zijn. Ook in de heuristische versie wordt, soms middels geboden hulp, de oplossing steeds gevonden. In geen enkel geval hoeft de computer het antwoord te geven. Op de resultaten en de statistische verwerking daarvan zal wellicht mettertijd in een apart artikel in dit tijdschrift worden ingegaan.

Noten

1. Het is niet persé noodzakelijk dat heuristische oplossingsmethoden volgens een 'discovery'-procedure en algoritmische oplossingsmethoden volgens een expositorische methode worden aangeleerd. Landa (1969) hoofdstuk 6 noemt bijv. de mogelijkheid om het algoritme door de leerling te laten ontdekken en zelfstandig te laten opstellen.
2. Hij gebruikt de term 'didactic teaching' voor de onderwijsmethode waarbij de leerkracht de begrippen, regels en wetmatigheden kant en klaar aan de leerling presenteert.
3. Het is in principe mogelijk de leerling zijn gang door de leerstof nagenoeg volledig zelf te laten bepalen. Dit is van belang als de leerling zich de structuur van een kennisgebied moet eigen maken. Vergelijking van het effect van zo'n leerling-gestuurde vorm van onderwijs met de meer 'traditionele' vorm van programma-gestuurd onderwijs heeft plaats in het onderzoek van G. Bernaert binnen het genoemde S.V.O.-C.A.I.-project (zie voortgangsrapporten S.V.O.).
4. Deze twijfel over de werkbaarheid van de Repetitor I wordt o.a. gedeeld door prof. Helmar Frank, Berlijn. Pogingen van zijn kant en van anderen om in Moskou het apparaat in werking te zien hadden helaas geen succes (persoonlijke mededeling).
5. Het hoeft niet zo te zijn dat a of b voor alle leerlingen een betere methode vormt. Het is mogelijk dat bepaalde leerstijlen van leerlingen zullen interacteren met deze twee mogelijke onderwijsprocedures ('treatments'). Uit de literatuur (Pask en Scott 1971 en 1972) is het onderscheid holistische-serialistische leerstijl bekend. Ruwweg corresponderen deze leerstijlen met voorkeur voor a of b. Zou een dergelijke 'aptitude-treatment' interactie kunnen worden vastgesteld dan zou de leerling op grond van deze kenmerken kunnen worden toegewezen aan één van deze mogelijke programmavormen.
De theoretische concepties van Pask en Scott zijn getoetst in het scriptie onderzoek van Beishuizen (1975). In het betreffende onderzoek wordt met een serialistische en een holistische versie van het algoritmische leerprogramma syllogismen-redeneringen gewerkt.

6. Ideaal is natuurlijk om op het betreffende continuüm een aantal punten te kiezen en hier onderwijsprogramma's bij te construeren. Omdat dit nog tijdrovender zou zijn dan het hier beschreven onderzoek al is hebben wij dit niet gedaan.
7. Bevindingen uit eigen onderzoek; ook Sand (Selz, 1935) beschrijft dit soort fouten als karakteristiek. Bijv. 2 5 9 14 —; karakteristieke fouten zijn 17 en 19.
8. Het zal duidelijk zijn dat een dergelijke multi-conditionele feedback alleen gegeven kan worden als van computer-faciliteiten gebruik kan worden gemaakt. Hierin schuilt één van de voordelen van C.A.I. boven conventionele geprogrammeerde instructie.

Literatuur

- Ausubel, D. P., *The Psychology of Meaningful Verbal Learning: An Introduction to School Learning*. New York, 1963
- Beishuizen, J. J., *Onderwijsstrategie en Leerstrategie; een Onderzoek naar de Interactie tussen Serialistische/Holistische Leerstrategie en Volgorde van Leerstof-aanbieding in een Algoritmisch Leerprogramma Syllogistisch Redeneren*. Doctoraalscriptie. Psych. Research Lab., V.U. 1975
- Bracht, G. H. en G. V. Glass, The external validity of experiments, *Am. Educ. Res. J.*, 1968, (5) 437-474
- Campbell, D. T. en J. C. Stanley, Experimental and Quasi-Experimental Designs for Research on Teaching, in: N. Gage (Ed.), *Handbook of Research on Teaching*. Chicago, 1963
- Cronbach, L. J., The Logic of Experiments on Discovery, in L. S. Schulman en E. R. Keislar (Eds.), *Learning by Discovery: A Critical Appraisal*. Chicago, 1966
- Cronbach, L. J. en R. E. Snow, *Individual Differences in Learning Ability as a Function of Instructional Variables* Final Report, Stanford Univ., Calif. School of Education, Stanford, 1969
- Elshout, M., *Onderzoek naar de Transferwaarde van Training in Cognitieve Vaardigheden*, Rapport Instituut voor Cognitie-Onderzoek, Universiteit van Amsterdam, ICO.089, 1973
- Guilford, J. P., *The Nature of Human Intelligence*, New York, 1967
- Hermans, H. J. M., *Prestatiemotief en Faalangst in Gezin en Onderwijs*. Amsterdam, 1971
- Kohnstamm, Ph., *Keur uit het Didactisch Werk*. Groningen, 1948
- Kuljutkin, Ju. N., Heuristische Methoden in het Oplossingsproces, 1970, samenvatting in: C. F. van Pareren en W. A. van Loon-Vervoorn: *Teksten en Analyses Sowjetpsychologie 1: Denken*. Groningen, 1975
- Landa, L. N., *Algorithmierung im Unterricht*. Berlin, 1969

- Landa, L. N., Instructional Grammar and Types of Thinking Activity, in: *Proceedings of the XVII International Congress of Applied Psychology*. Luik, 1971
- Leeuw, L. de, De Invloed van Associatieve Factoren en Conclusieinhoud op Syllogistisch Redeneren in Relatie tot twee Cognitieve Stijl Variabelen, *Nederlands Tijdschrift voor de Psychologie*, Deel XXVII, no. 11, 1973
- Leeuw, L. de, Aanleren van Algoritmische en Heuristische Oplossingsmethoden, in: *Empirische Studies over Onderwijs-18, Onderwijsresearchdag 1974*, Groningen, 1975
- Leeuw, L. de, Teaching of Algorithmic and Heuristic Problem Solving Methods by Computer-Assisted-Instruction, *Paper IFIP World Conference on Computers in Education*, Marseille, 1975.
- Miller, G. A., E. H. Galanter en H. H. Pribram, *Plans and the Structure of Behavior*. New York, 1960
- Parreren, C. F. van, *Psychologie van het Leren I*. Arnhem, 1969
- Parreren, C. F. en J. A. M. Carpay, *Sovjetpsychologen aan het Woord*. Groningen, 1972
- Parreren, C. F. en W. A. van Loon-Vervoorn, *Teksten en Analyses Sowjetpsychologen 1: Denken*. Groningen, 1975
- Pask, G. en B. C. E. Scott, Learning and Teaching Strategies in a Transformation Skill, *Br. J. Math. Statist. Psych*, 1971 (24) p. 205
- Pask, G. en B. C. E. Scott, Learning Strategies and Individual Competence, *Int. J. Man-Machine Studies*, 1972 (4) p. 217
- Selz, O., Versuche zur Hebung des Intelligenzniveaus. Ein Beitrag zur Theorie der Intelligenz und ihrer erzieherlichen Beeinflussung, *Zeitschrift für Psychologie*, 1935 (134).
- Wittrock, M. C., The Learning by Discovery Hypothesis, in: L. S. Shulman en E. R. Keislar (Eds.), *Learning by Discovery: A Critical Appraisal*. Chicago, 1966.

Curriculum vitae

L. de Leeuw (geb. 1936) studeerde psychologie aan de Vrije Universiteit van 1961 tot 1967, hoofdrichting Functieeler, Accent 'Leren en Denken'; tijdens de studie onderzoek op het gebied van de geprogrammeerde instructie.

Van 1967 tot 1969 werkzaam als wetenschappelijk medewerker bij het 'Nutsseminarium voor Pedagogiek' (het huidige 'Kohnstamm Instituut'). Van 1969 tot heden als wetenschappelijk medewerker verbonden aan het 'Psychologisch Research Laboratorium' van de Vrije Universiteit. Publiceerde o.m. over geprogrammeerde instructie en 'Computer-assisted-Instruction'.

Adres: De Boelelaan 1115, Amsterdam.