

LOGO: Achtergronden, evidentie en mogelijkheden

H. KUNST

Psychologisch Laboratorium, Rijksuniversiteit Utrecht

Samenvatting

In dit artikel wordt een overzicht gegeven van de programmeertaal LOGO en de onderwijsfilosofie erachter. Verder worden de psychologische onderbouwing en het evaluatieonderzoek besproken. Betoogd wordt, dat de claims ten aanzien van LOGO vaak niet al te sterk onderbouwd zijn, maar dat er niettemin in dit opzicht verbetering mogelijk is, terwijl LOGO bovendien aantrekkelijke mogelijkheden biedt om kinderen op een zinvolle manier met computers te laten omgaan. Tenslotte worden suggesties gedaan voor de mogelijke invoering van LOGO op scholen.

1 Inleiding

De computer en wat daar bij hoort verschijnt op drie manieren in het onderwijs: 1. als leerinhoud (informatica-w.o. leren programmeren, burgerinformatica-zie de bijdrage van Plomp in dit nummer), 2. als leermiddel (o.a. CAI), 3. als hulpmiddel bij de administratie. LOGO, het onderwerp van deze bijdrage, vormt in deze indeling een speciaal geval: het is zowel een leerinhoud, een taal waarin kinderen van verschillende leeftijden (en volwassenen) kunnen leren programmeren als een leermiddel met behulp waarvan men wiskundige begrippen en algemene probleemoplossingsvaardigheden zou kunnen verwerven. LOGO verdient echter vooral bijzondere aandacht omdat het in tegenstelling tot andere toepassingen van computers in het onderwijs, is opgezet vanuit een uitdagende visie op het hele onderwijs in het informatietijdperk.

2 Onderwijsopvatting en programmeertaal

LOGO is een interactieve¹ programmeertaal,

onwikkeld als leeromgeving voor jonge kinderen. In ruimere zin is het ook de naam voor de erachter liggende onderwijsfilosofie die door initiator Papert op meeslepende wijze wordt uiteengezet in zijn boek 'Mindstorms' (1980). Kort samengevat komt Paperts visie op het volgende neer.

Kennis wordt door het kind geconstrueerd, in de meest spectaculaire gevallen zonder veel formele instructie maar met gebruikmaking van het in de culturele omgeving aanwezige materiaal. In de omgeving waarin de meeste kinderen opgroeien is de wiskunde slecht vertegenwoordigd. Er is weliswaar overvloedig concreet materiaal beschikbaar voor het tot stand komen van een intuïtief getalbegrip (b.v. het in paren voorkomen van schoenen, sokken, vaders en moeders, messen en vorken, vgl. Papert, 1980, p. 7), maar over het algemeen is de wiskunde onderdeel van een vreemde bèta-cultuur, sterk gescheiden van de vertrouwde, vaak actief anti-wiskundige cultuur en doorgaans in scholen van buitenaf opgelegd.

Fundamentele veranderingen in de ontwikkeling van het denken zouden kunnen worden teweeggebracht door de cultuur waarin kinderen opgroeien te veranderen en te verrijken. Zo kunnen wiskundig en procedureel denken worden bevorderd door kinderen in een omgeving te plaatsen waarin wiskundige begrippen en systematische procedures in concrete vorm aanwezig zijn, zodat ze die op dezelfde wijze experimenterend zich eigen kunnen maken als b.v. de moedertaal in de 'gewone' omgeving. Op deze wijze zou op den duur de scheiding tussen alfa en bètacultuur kunnen verdwijnen, evenals de bij de meeste mensen bestaande wiskundevrees.

Met behulp van computers kan men de omgeving creëren waarin het kind met procedures en wiskundige begrippen als concrete objecten kan omgaan. LOGO als programmeertaal is niet bedoeld als eindproduct maar als prototype voor een dergelijke omgeving. Het romantische ideaal dat Papert voor ogen staat, is dat van kleine, informele gemeenschappen, gericht op een of andere gemeenschappelijke activiteit, waarin leren als min of meer natuurlijk neveneffect optreedt, waarbij volwassenen als adviseurs en medeleerlingen optreden en

die de scholen in de gebruikelijke zin zouden moeten vervangen (Papert, 1980, pp. 177-183).

Bij het ontwikkelen van de programmeertaal LOGO is uitgegaan van de opvatting dat men kinderen de meest geavanceerde instrumenten in handen moet geven en daarbij niet te veel moet letten op hardwarekosten, die immers snel dalen. Papert zet zich nadrukkelijk af tegen het gebruik van BASIC, waarvan vrijwel de enige verdienste is, dat het op de goedkoopste microcomputers past (Papert, 1980, pp. 33-

36).

Hoewel in eerste instantie ontwikkeld voor jonge kinderen is LOGO een volwaardige programmeertaal, die ook voor oudere kinderen en volwassenen ruimschoots intellectuele uitdagingen biedt (zie b.v. de procedures SUPERDRIEHOEK, C, OMGEKEERDE en GGD in Tabel 1, alsmede de Figuren 4 en 5).

LOGO is in vele opzichten door LISP geïnspireerd en onderscheidt zich daarmee van andere talen voor onderwijsdoeleinden die meer met BASIC, PASCAL of ALGOL gemeen

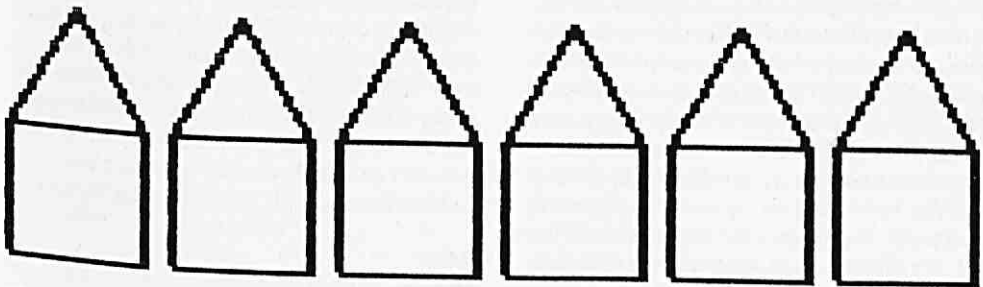
Tabel 1 Enige LOGO-procedures²

<pre>TO DRIEHOEK :L REPEAT 3 [FD :L RT 120] END TO VIERKANT :L REPEAT 4 [FD :L RT 120] END TO STRAAT :L :N REPEAT :N [HUIS :L PU RT 90 FD :L*1.2 PD LT 90] END TO CIRKEL FD 1 RT 1 IF HEADING = 0 THEN STOP CIRKEL END TO SLAK :N IF :N = 0 THEN STOP DRIEHOEK :N RT 12 SLAK :N-1 END TO SUPERDRIEHOEK :N :L IF :N = 0 THEN STOP REPEAT 3 [SUPERDRIEHOEK :N-1 :L/2 FD :L RT 120] END TO OMGEKEERDE :LIJST IF :LIJST = [] THEN OUTPUT :LIJST OUTPUT LPUT (FIRST :LIJST) (OMGEKEERDE BUTFIRST :LIJST) END TO GGD :M :N IF :N = 0 THEN OUTPUT :M OUTPUT GGD :N (REMAINDER :M :N) END</pre>	<pre>TO HUIS :L LT 90 DRIEHOEK :L FD :L RT 180 VIERKANT :L FD :L LT 90 END TO RVEELHOEK :N :L REPEAT :N [FD :L RT 360/:N] END TO C :N :L IF :N = 0 THEN FD :L STOP C :N-1 :L LT 90 C :N-1 :L RT 90 END</pre>
--	--

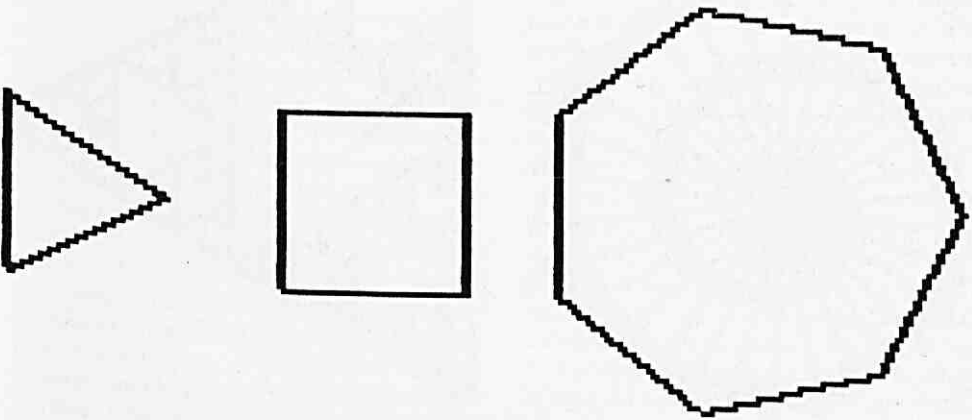
hebben³. Het programmeren in LOGO komt neer op het definiëren van procedures (zoals DRIEHOEK en VIERKANT in Tabel 1), die elk zelfstandig kunnen worden toegepast (d.w.z. zonder dat ze in een programma zijn opgenomen) en die op dezelfde wijze als elementaire instructies als bouwstenen voor andere procedures gebruikt kunnen worden (zie het gebruik van DRIEHOEK en VIERKANT in HUIS en SLAK en het gebruik van HUIS in STRAAT). Hierdoor wordt een modulaire programmeerstijl aangemoedigd. Programma's en data vertonen niet zo'n volledige overeenkomst in structuur als dat bij LISP het geval is, maar niettemin bevat LOGO de mogelijkheden tot lijstverwerking en kunnen programma's als lijsten worden weergegeven en verwerkt⁴. Recursie⁵ is in LOGO even vanzelfsprekend als in LISP (zie SLAK, C, SUPERDRIEHOEK, OMGEKEERDE en GGD in Tabel 1, alsmede Figuren 3-5) en wordt in handleidingen (b.v. Abelson, 1982) uitbundig gepropageerd.

Het bekendste onderdeel van LOGO vormen de instructies aan de schildpad ('turtle', hoewel het duidelijk om een landbewoner gaat), die in twee soorten voorkomt: een robotje dat eventueel een inktspoor achterlaat waarmee tekeningen kunnen worden gemaakt en een bewegend driehoekje dat eveneens een spoor achterlaat, maar dan op een beeldscherm. FORWARD 73 (afgekort FD 73) laat de schildpad 73 elementaire stapjes vooruit gaan, RIGHT 90 (RT 90) zorgt voor een kloksgewijze rotatie van 90°, LEFT 60 (LT 60) voor een rotatie van 60° in tegenovergestelde richting. PENUP (PU) en PENDOWN (PD) zorgen voor het achterwege laten en weer beginnen van een inktspoor.

Met behulp van de schildpad kan de beginner van meet af aan interessante resultaten bereiken (Figuren 1-3). Tegelijkertijd krijgt hij een aanknopingspunt voor het opbouwen van wiskundige begrippen. Figuren worden gedefinieerd als procedures die aansluiten bij de reeksen activiteiten die de kinderen zelf verrichten



Figuur 1 Output van STRAAT 30 6



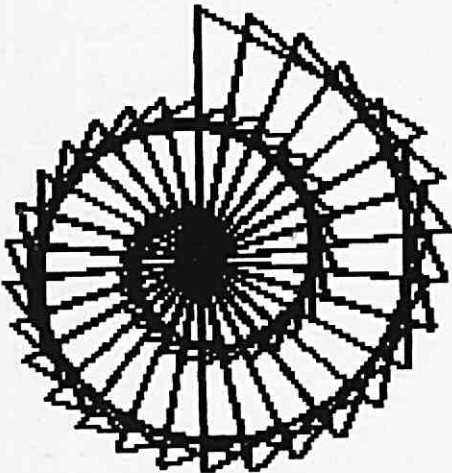
Figuur 2 Output van RVEELHOEK 3 40, RVEELHOEK 4 40, RVEELHOEK 7 40

wanneer ze figuren lopen of tekenen. De schildpadmeetkunde vormt een alternatief voor de analytische meetkunde (voor de manier waarop men via vectoranalyse een brug kan slaan, zie Abelson en Di Sessa 1981). Het definiëren van figuren in termen van lokale veranderingen (b.v. een cirkel als een curve met constante kromming) kan bovendien een startpunt vormen voor begrip van de differentiaal- en integraalrekening (Papert, 1980, p. 66). Hoe ver men met de turtlemeetkunde kan komen wordt gedemonstreerd in het wiskunde-curriculum voor highschool- en collegeleerlingen van Abelson en Di Sessa (1981), waarin o.a. vectoren, topologische onderwerpen en zelfs de algemene relativiteitstheorie worden behandeld. Andere 'powerful ideas' uit de wiskunde die in LOGO kunnen worden concreetiseerd zijn variabele, functie en, zoals reeds opgemerkt, recursie. De veronderstelde mogelijkheden van LOGO zijn niet beperkt tot de wiskunde maar doen zich ook voor in de fysica (Papert, 1980, pp. 48-50), talen (Böcker & Fischer, 1977) en muziekonderwijs (Pinxteren, 1982). Naast vakspecifieke begrippen en methoden biedt LOGO voorbeelden van zeer algemene methoden van leren en probleemoplossen. Met name het plannen van een oplossing door een probleem in deelproblemen te splitsen en het leren door fouten in procedures te verbeteren i.p.v. de gevolgde aanpak integraal te verwerpen en opnieuw te beginnen, worden al programmerend vrijwel vanzelf beoefend. LOGO en de schildpadmeetkunde in het bijzonder, zou een ideaal domein vormen

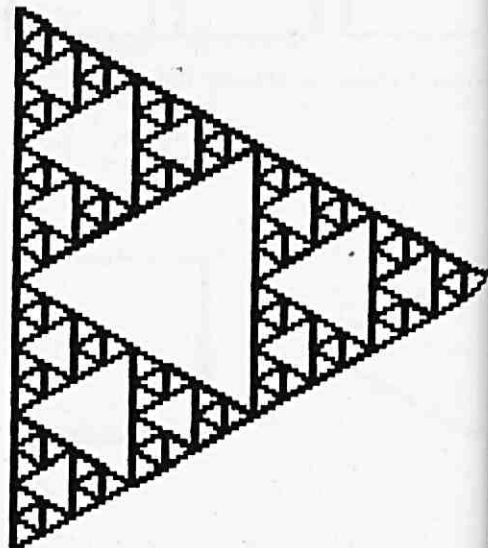
voor het in concrete vorm demonstreren van Polya's heuristische regels (zoals 'maak een plan', 'ken je een soortgelijk probleem', zie Polya, 1948; Papert, 1980, p. 64). Speciale aandacht verdient het werk met gehandicapte leerlingen waarvan gunstige resultaten worden gemeld (Weir, 1981).

Rond LOGO en de eraan ten grondslag liggende ideeën is een internationaal netwerk van onderzoeksgroepen en 'congregaties van gelovigen' ontstaan, veelal voorzien van eigen clubblaadjes. Een uitstekende indruk kan men krijgen uit het speciaal aan LOGO gewijde augustusnummer 1982 van *BYTE*. De belangrijkste onderzoekscentra zijn te vinden aan het MIT (Cambridge, Mass., USA), waar Papert hoogleraar is en de Universiteit van Edinburgh, waar een aanmerkelijk minder hemelbestormende aanpak wordt gevolgd: meer cursorisch onderwijs aan de hand van werkbladen (zie b.v. Howe, O'Shea & Plane, 1980). De bekendste van de scholen waar LOGO is ingevoerd is de Lamplighterschool te Dallas, waar eerste- t/m vierde klassers intensief met LOGO worden geconfronteerd (Gorman, 1981 a, b; Lamplighter Teachers, 1981). Veel publiciteit kreeg het Parijse Centre Mondial, waar Frankrijk door o.m. massale invoering van LOGO in het onderwijs hoog in de vaart der volkeren moet worden opgestoten, maar waar Papert inmiddels na een ernstig conflict is vertrokken.

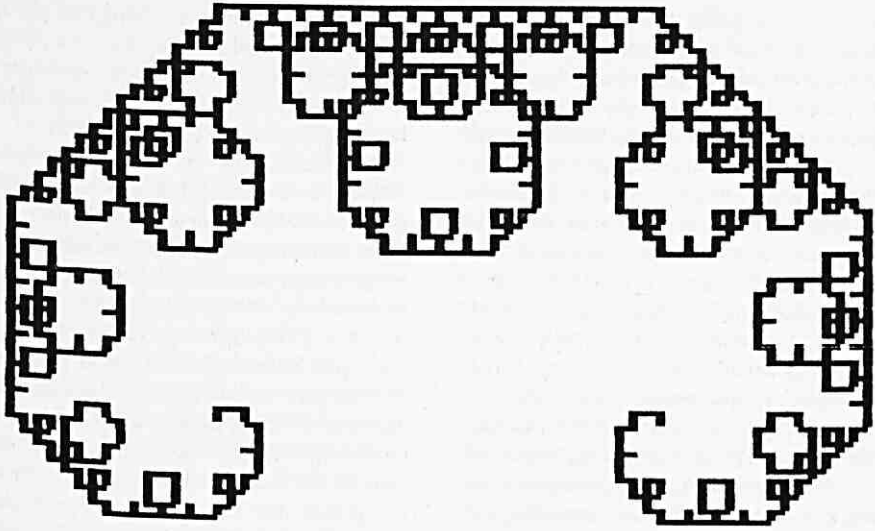
In Nederland is sinds kort het LOGO-centrum



Figuur 3 Output van SLAK 70



Figuur 4 Output van SUPERDRIEHOEK 5 150



Figuur 5 Output van C 10 3.

Nederland te Nijmegen⁶ (Pinxteren, Hoenkamp & Ringelberg, 1983 a, b) zeer actief bezig met de promotie van LOGO en het opzetten van proefprojecten – in eerste instantie ‘vacantiescholen’. In Utrecht wordt sinds kort op bescheiden schaal onderzoek gedaan naar het leren programmeren in LOGO bij brugklasleerlingen van MAVO-niveau (Janszen, 1983; Van Ree, 1983).

3 Psychologische fundering

In hoeverre zijn er vanuit de psychologie aanknopingspunten te vinden voor de grondgedachten achter LOGO, de wijze waarop deze worden uitgewerkt en de verwachtingen die ten aanzien van de effecten worden gewekt? Paperts visie op het kind als constructeur van zijn eigen intellectuele structuren, is expliciet gebaseerd op Piaget. ‘Piagetian learning’ is de term die Papert (1980, pp. 31-32) hanteert voor het ontwikkelen van kennis van o.a. de moedertaal en de schildpadmeetkunde door spontaan te opereren op de omgeving. De veel aangehaalde analogie tussen het leren van de moedertaal enerzijds en van de taal in Paperts ‘Mathland’ anderzijds, – treffende argument voor het curriculumloze leren –, berust op Piagets opvattingen over de relatie tussen taalontwikkeling en cognitieve ontwikkeling in het algemeen. Die zijn niet onomstreden (vgl. Chomsky 1980 – dat de strijd bepaald nog

niet in het voordeel van Piagets standpunt is beslecht, valt te constateren aan de hand van de discussies in Piattelli-Palmarini, 1979).

Paperts interpretatie van Piaget is sterk beïnvloed door de kunstmatige intelligentiebenaдерing die de laatste decennia binnen de cognitieve psychologie een steeds sterkere invloed is gaan uitoefenen. Piagets cognitieve structuren vat hij op als verzamelingen van min of meer zelfstandig opererende procedures, samenwerkend in een ‘society of mind’ (Papert, 1980, p. 169; Minsky, 1979). Het idee dat ons cognitieve apparaat is opgebouwd uit kleinere modules is overigens vrij populair in de cognitieve psychologie. De meest bekende formulering is dat van het productiesysteem (zie b.v. Newell & Simon, 1972; Anderson, 1980; Thibaut, Just & Carpenter, 1982). Optimisme over de leerbaarheid van modules van beperkte omvang ligt meer voor de hand dan over de leerbaarheid van alles bepalende globale structuren. Dat is dan ook een opvallend verschil tussen Papert en Piaget.

In de plaats van Piagets equilibratieproces komen bij Papert een drietal basismechanismen van leren en probleemoplossen, die niet in de laatste plaats ontleend zijn aan de praktijk van het programmeren en die dan ook in LOGO duidelijk geconcretiseerd worden. Dit zijn 1. het top-down opsplitsen van een probleem in deelproblemen; 2. ‘bricolage’, het al knutselend, bottom up samenstellen van nieuwe procedurele modules door coördinatie van ou-

dere en 3. 'debugging', het verwijderen van fouten uit procedures. Een cognitieve leertheorie waarin de onderlinge relaties tussen die mechanismen nader is uitgewerkt, bestaat momenteel niet. Opsplitsen in deelproblemen geldt vrij algemeen als een efficiënte probleemoplossingsmethode (Newell & Simon, 1972), niet in de laatste plaats bij het programmeren, waar een dergelijke aanpak niet alleen wordt aanbevolen, maar ook inderdaad de experts onderscheidt van de beginners (Jeffries, Atwood, Polson & Turner, 1982). Over de precieze mechanismen van bricolage is veel minder bekend. Suggesties voor mogelijke processen, verwerkt in nieuwe ideeën producerende computerprogramma's vindt men bij Lenat (1982). Iets soortgelijks geldt voor debugging. Een effectief zelflerend computerprogramma met debugging als voornaamste mechanisme is beschreven in Sussman (1975).

Papert kent een veel grotere invloed van de cultuur op de cognitieve ontwikkeling toe dan Piaget. De piagetiaanse stadia zijn volgens hem geen gevolg van een of andere natuurlijke ontwikkeling, maar van de pre-computer-cultuur waarin de huidige Westerlingen (meestal nog) leven, evenals de Afrikaanse volkeren, waaraan de universaliteit van de Piagetiaanse stadia wordt getoetst. Met name het late bereiken van het zgn. formele stadium zou in een echte computercultuur niet nodig zijn – het zou zelfs voorstelbaar zijn dat het inzicht in procedures, dat kenmerkend is voor het formele stadium, bereikt wordt vóór het kwantitatieve inzicht van het concrete stadium (Papert, 1980, pp. 174-176). Het grote belang dat Papert hecht aan de cultuur en de in de samenleving aanwezige werktuigen als doorslaggevend factoren in de cognitieve ontwikkeling, brengen Papert naar mijn mening in zeker opzicht in de buurt van theoretici als Vygotskij en Gal'perin (Van Parreren & Carpay, 1980; Talyzina, 1980). LOGO zou kunnen worden opgevat als een alternatief voor de verbale fase in de opbouw van mentale handelingen: anders dan een verbale omschrijving kan een omschrijving in LOGO een 'gematerialiseerde' activiteit te weegbrengen die dicht bij materiële handelingen van het kind aansluiten. Zowel LOGO zelf als de bespreking van LOGO-programma's vergroten de mogelijkheden om ook op het eerste gezicht niet-formuleerbare intuïties in taal uit te drukken. Onderzoek afkomstig uit of geïnspireerd door de 'Russische school' onder-

steunt het belang van debugging als fundamenteel leermechanisme, b.v. het onderzoek naar de invloed van reflectie op de begripvorming (Zak, 1980) en van de foutenanalyse bij het motorisch leren (Pijning, 1975).

Afwijkend van de hierboven vermelde school van sovjetpsychologen is natuurlijk de grote mate waarin, in de door Papert voorgestane leerstuitatie, de controle aan de leerling wordt overgelaten. Het debat over de voor- en nadelen van zelf-ontdekkend leren is oud en onbeslist (Van Parreren, 1981a). De superioriteit van zelfontdekte kennis is geen onomstreden gegeven en de kans dat een leerling iets niet ontdekt moet worden afgewogen tegen de motivationele voordelen van het zelf-ontdeken. Er zijn vele vaardigheden en inzichten die bij gebrek aan een praktisch overdraagbare methode wel grotendeels zelf ontdekt moeten worden, maar juist Papert geeft voorbeelden van uitstekend overdraagbare procedures voor vaardigheden die op het eerste gezicht niet onder woorden te brengen zijn (b.v. jongleren, Papert, 1980, p.p. 105-113) en waar de meeste mensen niet gauw zelfstandig op zullen komen. Er zijn dan ook onderzoekscentra waar een veel meer voorgestructureerde aanpak van LOGO wordt voorgestaan, zoals het Department of Artificial Intelligence te Edinburgh.

Een onderneming als LOGO impliceert een groot optimisme over de mogelijkheden van transfer. 'Micro werelden' zoals de schildpadmeetkunde zijn volgens Papert broedmachines van 'powerful ideas', die door overbrenging naar andere microwerelden tot algemene principes kunnen uitgroeien. Een voorbeeld, waarin bij één kind door coördinatie van principes uit meerdere microwerelden (de combinatie van tientallen graden in een LOGO-spelletje, het omgaan met zakgeld en de schoolsommen) begrip voor optellen ontstaat, wordt uitgebreid geanalyseerd door Lawler (1981). Het voorbeeld suggereert dat het beschikbaar hebben van en het leggen van verbanden met situaties buiten LOGO een belangrijke factor is in het tot stand komen van algemeen begrip. Garantie dat het optreedt is er echter niet, laat staan dat het op een bepaald moment optreedt. Van het mechanisme dat de coördinatie veroorzaakt is nog maar weinig bekend.

Sinds Thorndike's onderzoek naar de heilzame effecten van het onderwijs in de klassieke talen (1924) is sceptis aangaande transfer ge-

bruikbaar. Hoewel het verschijnsel bestaat en er het een en ander bekend is over de omstandigheden waaronder het optreedt (Woodworth & Scholsberg, 1954 p.p. 733-778; Simon, 1980; Van Parreren, 1981b), is de fundamentele kennis over het leerproces te gebrekkig om daarover in specifieke gevallen op voorhand voorspellingen te doen. Onderzoek naar de effecten van leren programmeren is nog schaars en de resultaten zijn wisselend (Seidman 1982; Soloway, Lochhead & Clement 1982; Dirkzwager in voorbereiding). In afwachting van meer onderzoek kan veilig worden gezegd dat de eventuele transfer-effecten van LOGO vooral zullen optreden wanneer voor de leerling expliciete verbanden worden gelegd met situaties buiten de LOGO-micro-wereld.

Vanuit de psychologie van het leren programmeren (zie b.v. Schneiderman, 1980) zijn nog niet veel argumenten voor of tegen LOGO te vinden. Het onderzoek naar de effecten van kenmerken van programmeertalen op het leren programmeren is niet zo relevant. Het merendeel van het onderzoek heeft betrekking op zeer beperkte programmeertalen waarin zeer weinig geoefend wordt (Sheil, 1981). De argumenten tegen het gebruik van een primitieve taal als BASIC (Papert, 1980, pp. 34-37) zijn zeer plausibel maar worden (nog) niet ondersteund door empirische gegevens. Hetzelfde geldt voor Dijkstra's mening dat het met aspirant informatici die ooit met BASIC zijn begonnen nooit meer goed komt (Dijkstra, 1982, p. 130). Er is enige evidentie voor de opvatting dat beginners minder moeite hebben met hogere orde talen dan met lagere orde talen (Du Boulay & O'Shea, 1980, zij het dat de gebruikte lagere ordetaal wel erg dicht bij een machinetaal stond). Daarnaast is het niet zeker dat ze geavanceerde constructen werkelijk begrijpen – ook wanneer ze die toepassen (Soloway et al., 1982, p. 183). In zijn experimenten met SOLO – een op LOGO lijkende kunstmatige intelligentietaal voor cursisten aan de Open University – vond Kahney (1982) dat na behandeling van recursieve programma's de cursisten doorgaans òf in het geheel geen consistent mentaal model van recursie bezaten òf een inadequaat loop-model. Mijn eigen ervaringen met een LISP-cursus aan psychologie-studenten komen hiermee overeen. Overigens wordt het verwarren van recursie en iteratie⁷ m.i. in de hand gewerkt door de gewoonte

recursie in LOGO (èn SOLO) te introduceren aan de hand van eenvoudige staartrecursie (Papert, 1980 p. 7; Abelson, 1982, p. 33; McDougall, Adams & Adams, 1982, p. 79; Kahney, 1982, p. 1-18). Claims dat kinderen in een LOGO-omgeving recursie onder de knie krijgen dienen te worden ondersteund met méér evidentie dan de trotse presentatie van enige door hen geschreven procedures.

Hoewel LOGO niet op een bijzonder degelijke psychologische ondergrond gefundeerd is, moet niet uit het oog worden verloren dat veel van de erachter liggende ideeën binnen de psychologie vrij algemeen onderschreven worden, zoals het opvatten van kennis als (afgeleid van) operaties (procedures, handelingsstructuren), het belang van eigen activiteit voor het verwerven van kennis, het belang van het expliciteren (waar mogelijk) van probleemoplossingsmethoden. Bovendien zijn er binnen de psychologie veel aanknopingspunten voor het alsnog leggen van een dergelijke ondergrond.

4 Evaluatie

Er zijn velerlei doelstellingen die met LOGO kunnen worden nagestreefd. Men kan ze in drie categorieën indelen. Ten eerste het leren programmeren, met alle verfijningen die men daarin kan aanbrengen. Ten tweede de bredere transfereffecten, de algemene en meer specifieke 'powerful ideas', de beïnvloeding van de ontwikkeling van het denken, de positieve gevolgen voor het onderwijs in wiskunde en andere vakken. Tenslotte zijn er de effecten op attitudes van leerlingen zoals een positievere houding ten opzichte van computers, wiskunde, leren in het algemeen of het oplossen van problemen.

Ten aanzien van de in de onderwijsresearch gangbare wijzen van evaluatie stelt Papert zich negatief op. Hij verwijst naar het verhaal van de negentien-eeuwse onderzoeker die wilde nagaan of machines beter waren dan paarden, naast vier paarden een 1 pk machine inspande en na een half jaar statistische berekeningen tot de conclusie kwam dat er sprake was van een Hawthorne-effect op de paarden (Papert, 1971). Effecten die men door een statisticus moet laten controleren zijn ipso facto niet interessant (Papert, 1973). Deze tirades tegen de zeden en gewoonten in de onderwijsresearch zijn niet geheel ongegrond (zie ook Freud-

thal, 1980, p. 144 e.v.). De kolossale effecten die wél interessant zouden zijn bleven vooralsnog beperkt tot het leren programmeren – waar een vergelijking met leerlingen die niet leren programmeren al gauw een kolossaal effect oplevert.

Evaluatieonderzoek met betrekking tot LOGO is vrij schaars – zeker in vergelijking met de vele stukken waarin de verdiensten van LOGO luidruchtig worden uitgedragen. Het is bovendien verspreid in interne rapporten en clubblaadjes. Een overzicht treft men aan bij Pinxteren et al. (1983b).

Een voorbeeld van een uitvoerige kwalitatieve evaluatie vindt men in Papert, Watt, Di Sessa & Weir (1981). Daarin staat een uitvoerige analyse van het werk van 16 zesdeklassers die gedurende één schooljaar 20-40 uur met LOGO werkten. Op twee na leerden zij zelfstandig programma's te schrijven, maar welke methoden en concepten zij onder de knie kregen, liep sterk uiteen. Dit werd niet in de laatste plaats veroorzaakt door de verscheidenheid van projecten waaraan werd gewerkt. Diverse projecten waren overigens van een opmerkelijk niveau. Het is vaak moeilijk vast te stellen in hoeverre het verkregen inzicht zich uitstrekt buiten de geproduceerde programma's en in hoeverre die op grond van correct begrip zijn gemaakt. De voorbeelden van recursieve programma's betroffen vrijwel alleen simpele staartrecursie die – zoals hierboven betoogd – ook op basis van een inadequaat model geschreven kunnen worden. Niettemin kan men op grond van dit rapport en van andere (b.v. Pinxteren et al., 1983b) stellen dat de leerlingen uit deze leeftijdsgroep in een betrekkelijk ongestructureerde leersituatie zowel van programmeren als van schildpadmeetkunde het nodige opsteken.

Transfereffecten zijn veel moeilijker vast te stellen. De transfertaken van Papert et al. waren niet bijzonder opwindend (hoeken schatten, lengtes van lijnen schatten, reeksen stappen vooruit en achteruit in één stap uitdrukken, een route door een rechthoekig stratennetwerk beschrijven) en de resultaten waren zelfs niet significant. In het onderzoek van Howe et al. (1980), waarin dichter werd aangesloten bij de door Papert zo verfoeide onderzoeksmethoden, werden de effecten op de wiskundeprestaties nagegaan van een veel sterker voorgestructureerde tweejarige LOGO cursus in de eerste twee jaren van het voortgezet onderwijs.

Er werden enige bescheiden positieve effecten gevonden. Eén resultaat uit dit onderzoek verdient bijzondere aandacht. Uit vragenlijsten die de wiskundeleerkrachten over elk van de leerlingen invulden, bleek dat de experimentele groep volgens hen veel verstandiger over wiskundige onderwerpen kon spreken en hun moeilijkheden beter onder woorden kon brengen. Informele observaties van buitenstaanders wezen in dezelfde richting. Een dergelijk effect, als het door grondiger onderzoek bevestigd kan worden, is wellicht belangrijker dan een rechtstreekse verbetering in de wiskundeprestaties. Het zou een hoger rendement voor later onderwijs kunnen betekenen.

In het algemeen lijken de transfereffecten van LOGO enigszins tegenstrijdig en vooralsnog niet spectaculair te zijn (Pinxteren et al. 1983b)⁸.

Wat betreft de attitudes van leerlingen moet men, zelfs wanneer men de vaak propagandistisch gestelde publikaties (b.v. Lamplighter teachers, 1981; Gorman, 1981 a, b) en de informele observaties met enige korrels zout neemt, tot de conclusie komen dat leerlingen doorgaans met plezier bezig zijn (Papert et al. 1979; Pinxteren et al. 1983b). Dat is vooral het geval wanneer de gelegenheid bestaat samen te werken (Janszen, 1983). In sommige gevallen treedt overigens na een enthousiast begin een zekere terugval op (Howe et al., 1980). Groter zelfvertrouwen ten opzichte van wiskundige onderwerpen wordt in anecdotes van diverse auteurs vermeld.

Het evaluatieonderzoek rond LOGO laat nog veel te wensen over. Bij de kwalitatieve beschrijvingen à la Papert et al. (1980) zijn de taken te weinig omschreven. Terwijl anderzijds bij het toepassen van gebruikelijke onderzoeksdesigns (b.v. Howe et al., 1980) globale prestatie-maten worden gebruikt, waarin de eventueel opgedane inzichten en vaardigheden slecht tot uiting kunnen komen. Met behulp van protocolanalyse aan de hand van standaardopgaven, het laten voorspellen van de resultaten van procedures en mogelijk ook geheugenexperimenten (voor goede voorbeelden, zie Kahnney, 1982) kan men de verworven vaardigheden beter in kaart brengen.

Belangrijker is nog, dat uitstraling naar andere terreinen vooral dan gerealiseerd wordt, wanneer die op die andere terreinen ook wordt bevorderd. Dat betekent bijvoorbeeld dat bij het onderwijs in wiskunde en andere vakken

expliciet wordt verwezen naar de in de LOGO-omgeving verworven kennis. Ook dan zijn de omstandigheden voor een optimaal effect nog niet per se gunstig. Wanneer inzichten en begrippen worden verworven, waarvan elders geen gebruik wordt gemaakt, zal men uiteraard weinig transfer kunnen verwachten. Veel van de stof uit een leergang als 'Moderne Wetkunde' (ook gebruikt door de Schotse leerlingen in het onderzoek van Howe et al., (1980) kan aan de hand van LOGO-procedures worden verduidelijkt, maar men zou vermoedelijk veel beter andere onderwerpen kunnen behandelen (b.v. Abelson & Di Sessa, 1981). Wat geschikt geacht wordt voor leerlingen wordt immers evenzeer bepaald door wat ze geacht worden niet te weten als door wat ze geacht worden wel te weten, zodat voor onverwachte inzichten in de schoolstof vaak geen emplot is. De drastische ingrepen in het onderwijs die nodig zouden zijn om Paperts verder gaande claims te toetsen zijn in een onderzoek moeilijk te realiseren.

Tenslotte is er nog het probleem van de controlegroep. Waartegen moet men de effecten afzetten? Tegen het gewone onderwijs, waaraan altijd wel wat te verbeteren valt, of tegen anderssoortige pogingen om sterk verwante doelstellingen te realiseren (b.v. Freudenthal, 1980)? Wanneer bovendien verschillende inhoud worden aangeleerd, wordt de vergelijking wel heel moeilijk. Beter is misschien Hofstee's aanbevelingen omtrent evaluatie serieus te nemen (Hofstee, 1982) en weddenschappen af te sluiten over omschreven effecten. Bijvoorbeeld (om te beginnen): na één schooljaar één lesuur per week een min of meer voorgestructureerde omgang met LOGO te hebben gehad, zullen eerste klas MAVO-leerlingen via een topdown aanpak een complex schildpadmeetkundeprogramma kunnen schrijven (b.v. een programma dat een locomotief met n wagons tekent) en zullen zij recursieve programma's (niet alleen staartrecursie) kunnen schrijven en begrijpen.

Hoewel voorlopig een zekere scepsis tegenover vooral de verstrekkende claims ten aanzien van LOGO gewettigd is, moet niet uit het oog worden verloren dat over LOGO als programmeertaal voor leerlingen beneden de 15 jaar méér bekend is dan over enige programmeertaal en dat het voor kinderen van diverse leeftijden mogelijk is om via LOGO op een mogelijk beperkte, maar in ieder geval zinnige

en plezierige manier met computers om te gaan.

5 LOGO en de scholen

De discussie rond de invoering van informatica op scholen concentreert zich vooral op het voortgezet onderwijs. Ten aanzien van het basisonderwijs wordt opgemerkt dat kinderen daar weliswaar computers op niet-triviale wijze kunnen leren gebruiken, maar dat de gebruikelijke didactische werkvormen te zeer zouden moeten worden gewijzigd om verwezenlijking op korte termijn mogelijk te maken (Adviescommissie, 1982, p. 26). Men kan daarmee instemmen, met dien verstande dat LOGO een goed uitgangspunt biedt om die didactische werkvormen te ontwikkelen, of voor Nederlands gebruik te bewerken, en dat die niet-triviale wijze van omgaan met computers daar heel wat verder kan gaan dan 'elektronische spelletjes te spelen of vaardigheden met behulp van de computer te oefenen' (Adviescommissie, 1982, p. 17). Bovendien zouden nu althans pogingen gedaan moeten worden om invoering op langere termijn goed voor te bereiden door proefprojecten op de scholen en vooral PABO's.

Het huidige beleid is gericht op invoering op korte termijn van de informatica in de beginfase van het voortgezet onderwijs onder de naam 'burgerinformatica'. Nu is LOGO nooit bedoeld als vulling van een vak informatica, laat staan 'burgerinformatica'. Toch is er reden om de invoering van LOGO in het begin van het voortgezet onderwijs te overwegen. Zoals Uhlenbeck, De Bruijn en Levelt (1982) opmerken, kan men inzicht in de belangrijke rol die computers in de samenleving spelen inderdaad aanbrenge aan de hand van b.v. videomateriaal, waarvan de vertoning niet aan informaticaonderwijs gekoppeld hoeft te zijn. Een al te strikte koppeling, waarbij leerlingen b.v. op een microcomputer in BASIC de postgiro nabootsen, zou vermoedelijk eerder misleidend dan informatief zijn. Wel zal de vertrouwde uit eigen ervaring met programmeren en met algoritmen een grondiger behandeling van de rol van computers in de samenleving mogelijk maken. Bovendien is de redenering van Uhlenbeck et al. (1982), dat het behandelen van algoritmieken in de tweede helft van het voortgezet onderwijs de voorkeur ver-

dient, omdat dan gesteund wordt op enige jaren wiskundeonderwijs, omkeerbaar. Wiskundeonderwijs zou juist kunnen profiteren van LOGO. Tenslotte laten weliswaar de theoretische en empirische onderbouwing van LOGO nog veel te wensen over, maar als men invoering op korte termijn wil, is er nauwelijks een beter alternatief. Naar mijn mening is LOGO beter doordacht dan enige andere onderwijs-taal.

De volgende opzet zou m.i. serieuze overweging verdienen. Invoering van LOGO als min of meer zelfstandig onderdeel op basis van één à twee lessen per week. Op grond van o.a. de bevindingen van Howe et al. (1980) en onze eigen zeer recente ervaring met MAVO-leerlingen (Van Ree, 1983; Janszen, 1983; Koppen en Van Vuuren in voorbereiding) kan verwacht worden dat in één jaar het schrijven van programma's van enige complexiteit geleerd kan worden. In een tweede jaar zouden geavanceerde methoden behandeld kunnen worden en zelfstandig grotere projecten kunnen worden uitgevoerd. Tegelijkertijd zou in andere onderwijsonderdelen gebruik moeten worden gemaakt van de verworven inzichten. Dat geldt voor de behandeling van de rol van computers in de samenleving, maar ook voor diverse gangbare schoolvakken.

In het wiskundeonderwijs, maar ook bij natuurkunde, biologie en talen kan men bij de leerstofkeuze rekening houden met de kennis die die leerlingen bij het werken met LOGO hebben opgedaan. Men kan zelfs de leerlingen belangrijke begrippen als LOGO-procedures laten formuleren ten einde ze zich volledig eigen te laten maken.

Na de brugperiode (voortgezet basisonderwijs, middenschool) zou men een goede ondergrond hebben voor een meer gespecialiseerde behandeling van onderwerpen uit de informatica – b.v. tekstverwerking en COBOL-programmering voor het administratief onderwijs, theoretische informatica en desgewenst Pascal- of LISP-programmering in het VWO – ofschoon er eigenlijk niets tegen is gewoon met LOGO door te gaan.

Het is duidelijk dat deze voorstellen ver afstaan van de revolutionaire veranderingen die Papert op het oog heeft. De beschikbare gegevens rechtvaardigen echter niet meer dan een bescheiden aanzet, die op langere termijn tot radicale veranderingen aanleiding zou kunnen geven. Voor activiteiten buiten het onderwijs

om, in vakantiescholen, gebruikersclubs en 'dropincenters' blijft natuurlijk alle ruimte, al lijkt Paperts verwachting dat deze het reguliere onderwijs zullen gaan vervangen, voorlopig erg optimistisch, of zo men (niet) wil, pessimistisch.

Noten

1. Het is in deze taal mogelijk rechtstreeks te 'converseren' met de computer, doordat men direct antwoord krijgt op ingevoerde instructies en gegevens.
2. Het gebruikte dialect is Terrapin-LOGO, een van de dialecten voor de microcomputer Apple II. Toelichting voor zover niet in de tekst: een dubbele punt geeft aan dat de waarde wordt bedoeld die voor de desbetreffende variabele is ingevuld. TO definieert de procedure. FIRST geeft het eerste element van een lijst (zie noot 4), BUTFIRST geeft de rest van de lijst. LPUT plaatst het eerste argument achterin de lijst die door het tweede argument wordt aangeduid, OUTPUT stopt de procedure. De procedure krijgt nu de waarde van wat er achter OUTPUT vermeld staat. Bijv.: OMGEKEERDE [A B C] krijgt de waarde [C B A]; GGD 16 12 krijgt de waarde 4. De schildpad staat op het beeldscherm met een HEADING van 0, d.w.z. met z'n kop naar boven.
3. LISP (LISt Processing language) is een zeer flexibele taal voor symboolmanipulatie. Het is de taal die het meest gebruikt wordt voor kunstmatige intelligentie. Zowel de procedure als de gegevens waarop deze werken hebben een lijststructuur (zie noot 4).
4. Lijsten zijn flexibele datastructuren, bestaande uit een reeks elementen die elk zijn voorzien van een verwijzing naar een volgend element. Notatie bijvoorbeeld [AAP NOOT MIES]. De procedure OMGEKEERDE in Tabel 1 is een voorbeeld van een lijstverwerkende procedure. OMGEKEERDE [AAP NOOT MIES] = [MIES NOOT AAP]. De elementen kunnen zelf weer lijsten zijn bijv. in: [[: L] [REPEAT 3 [FD : L RT 120]]], een lijst die de definitie van de procedure DRIEHOEK in Tabel 1 weergeeft. Het is mogelijk om door lijstverwerkingsprocedures andere procedures te laten schrijven of wijzigen.
5. Recursie is het hanteren van een begrip in de definitie van een begrip of het aanroepen van een procedure in de definitie van die procedure. Recursie wordt fraai in beeld gebracht door de befaamde Droste-cacaobus. Wanneer een simpelste geval gedefinieerd is, of een stopconditie in de procedure is opgenomen kan recursie een elegante manier zijn om definities te geven of procedures te schrijven. Voorbeeld: 'Een superdrie-

hoek (zie Figuur 4) is een gelijkzijdige driehoek of een figuur, bestaande uit drie congruente superdriehoeken die elk met één van de beide andere superdriehoeken één hoekpunt gemeen heeft en met de andere superdriehoek een ander hoekpunt. Voor recursieve procedures zie Tabel 1 en vooral de procedure SUPERDRIEHOEK.

6. Canisiussingel 26, 6511 TJ Nijmegen.
7. Iteratie is het binnen een procedure een aantal malen herhalen van een aantal stappen. Een simpele vorm ervan vindt men in de REPEAT instructie in DRIEHOEK, VIERKANT en STRAAT (Tabel 1), waar het aantal iteraties van tevoren is opgegeven, maar dat aantal kan ook b.v. van de resultaten van de iteraties zelf afhangen. Het belangrijke verschil met recursie (zie noot 5) is, dat bij recursie steeds nieuwe versies van de procedures worden aangeroepen terwijl bij iteratie de herhaling steeds op hetzelfde niveau binnen de procedure plaatsvindt. Bij staartrecursie wordt één recursieve aanroep aan het eind van de procedure gedaan, om een in essentie iteratief proces te weeg te brengen. Bij CIRKEL heeft de recursieve aanroep aan het eind geen ander effect, dan dat FD 1 en RT 1 nogmaals gedaan worden, hetgeen net zo goed met een iteratieve loop gedaan zou kunnen worden. Wanneer geen andere vormen van recursie worden vertoond, zoals b.v. SUPERDRIEHOEK of C, die op iteratieve wijze nauwelijks te produceren zijn, kan het misverstand ontstaan dat een recursieve aanroep in feite het terugspringen naar de eerste stap van de procedure zou zijn.
8. Na het inleveren van het manuscript bereikte mij Gorman, H. & L. E. Bourne, Learning to think by learning in third-grade computer programmers. *Bulletin of the Psychonomic Society*, 1983, 21, 165-167, waarin een significant verschil in leerprestaties op een regel-leertaak wordt vastgesteld tussen leerlingen die één uur per week met LOGO werkten en leerlingen die een half per uur week met LOGO werkten.
9. Dit is een gissing: over de invulling van het vak burgerinformatica is mij nu (juli 1983) nog vrijwel niets bekend.

Literatuur

Abelson, H., *Apple logo*, Peterborough NH: Byte/McGraw Hill, 1982.

Abelson, H. & A. Di Sessa, *Turtle geometry*, Cambridge, Mass: MIT-Press, 1981.

Adviescommissie Onderwijs en Informatietechnologie, *Leren over informatietechnologie: Noodzaak voor iedereen.* 's-Gravenhage: Staatsuitgeverij, 1982.

Anderson, J. R., *Cognitive psychology and its implications*. San Francisco: Freeman, 1980.

Böcker, H. D. & G. Fischer, *Arbeitsmaterialien zum*

interaktiven Problemlösen mit Computerhilfe, Darmstadt: Hessisches Institut für Bildungsplanung und Schulentwicklung, 1977.

Chomsky, N., *Rules and representations*, New York: Columbia University Press, 1980.

Dijkstra, E. W., *Selected writings on computing: A personal perspective*. New York: Springer, 1982.

Dirkzwager, A., *Leren programmeren op de basisschool. Rapport in voorbereiding*.

Du Boulay, B. & T. O'Shea, Teaching novices programming. In: M. Coombs (Ed.), *Computing skills and adaptive systems*. New York: Academic Press, 1980.

Freudenthal, H., *Weeding and sowing: Preface to a science of mathematical education, 2e druk*. Dordrecht: Reidel 1980.

Gorman, H., The lamplighter LOGO project. *99 er Magazine*, 1981, 1, 60-62.

Gorman, H., Logo and the space shuttle: An update on lamplighter activities. *99 er Magazine*, 1981, 1, 64-65.

Hofstee, W. K. B., Evaluatie: Een methodologische analyse. *Tijdschrift voor Onderwijsresearch*, 1982, 7, 193-202.

Howe, J. A. M., T. O'Shea, & F. Plane, Teaching Mathematics through logo: An evaluation study. In: R. Lewis & W. Tagg, (Eds), *Computer assisted learning*. Amsterdam: North Holland, 1980.

Janszen, H., *Het Utrechtse logoproject*, stageverslag R. U. Utrecht 1983.

Jeffries, R., A. A. Turner, P. G. Polson, & M. E. Atwood, The processes involved in designing software. In: J. R. Anderson, (Ed.), *Cognitive skills and their acquisition*. Hillsdale N. J.: Erlbaum, 1981.

Kahney, H., *An in depth study of the behavior of novice programmers*. Technical Report No 82-9. Human Cognition Research Group, The Open University Milton Keynes, 1982.

Koppen, M. & J. van Vuuren, Stageverslag in voorbereiding. Lamplighter, Learning at the lamplighter school. *Microcomputing*, 1981, V, 42-47.

Lawler, B., The progressive construction of mind. *Cognitive Science*, 1981, 5, 1-30.

Lenat, D., The nature of heuristics. *Artificial Intelligence*, 1982, 9, 189-249.

McDougall, A., T. Adams, & P. Adams, *Learning logo on the apple*. Sydney: Prentice Hall of Australia, 1982.

Minsky, M., The society theory of thinking. In: P. H. Winston & R. H. Brown (Eds), *Artificial Intelligence: An MIT perspective, vol 1*, Cambridge, Mass.: MIT-press, 1979.

Newell, A. & H. A. Simon, *Human problem solving*, Englewood Cliffs N. J.: Prentice Hall, 1972.

Papert, S., *Teaching children thinking*, Logo Memo 2, MIT, 1971.

Papert, S., *Uses of technology to enhance education*. Logo Memo 8, MIT, 1973.

Papert, S., *Mindstorms. Children, computers and po-*

- werful ideas. New York: Basic Books, 1980.
- Papert, S., D. Watt, A. di Sessa, & S. Weir, *Final report of the Brookline logo project part II: Project summary and data analysis*. Logo Memo 53, MIT, 1973.
- Parreren, C. F. van, *Onderwijspsychologie*. In: C. F. van Parreren & M. C. Schouten-van Parreren, *Onderwijsproceskunde*. Groningen: Wolters-Noordhoff, 1981a.
- Parreren, C. F. van, *Systeemscheiding, Verschijnen en theorie*. In: C. F. van Parreren & M. C. Schouten-van Parreren, *Onderwijsproceskunde*. Groningen: Wolters-Noordhoff, 1981b.
- Parreren, C. F. van & J. A. M. Carpay, *Soviet psychologen over onderwijs en cognitieve ontwikkeling*. Groningen: Wolters-Noordhoff, 1980.
- Pijning, H. F., *Leren van een groot-motorische vaardigheid: Een kwalitatief leerpsychologisch onderzoek van de onderhandse volleybalservice*. Dissertatie Rijksuniversiteit Utrecht, 1975.
- Pinxteren, H., *Logo muziek 1*. Logo Memo 8, Logo Centrum Nederland, 1983.
- Pinxteren, H., E. Hoenkamp & J. Ringelberg, *T.I.-logo*. Logo Memo 6, Logo Centrum Nederland, 1983a.
- Pinxteren, H., E. Hoenkamp & J. Ringelberg, *Logo: Een evaluatie Logo Notitie 3*. Logo Centrum Nederland, 1983b.
- Polya, G., *How to solve it, a new aspect of mathematical method* 5th print. Princeton N.J.: Princeton University Press, 1948.
- Ree, H. van, *Stageverslag*, RUU, 1983.
- Seidman, R. H., *The effects of learning the logo computer programming language on conditional reasoning in school children*. University Microfilms International, Ann Arbor: 1982.
- Schneiderman, B., *Software Psychology: Human factors in computers and information systems*. Cambridge Mass.: Winthrop, 1980.
- Sessa, A. di, *Unlearning Aristotelian physics: A study of knowledge based learning*. *Cognitive Science*, 1982, 6, 37-75.
- Sheil, B. A., *The psychological study of programming*. *ACM Computing Surveys* 1981, 13, 101-120.
- Simon, H. A., *Problem solving and education*. In: D. T. Tuma & F. Reif, *Problem solving and education: Issues in teaching and research*. Hillsdale NJ: Erlbaum, 1980.
- Soloway, E., J. Lockhead & J. Clement, *Does computer programming enhance problem solving ability? Some positive evidence on algebra word problems*. In: R. J. Seidel, R. E. Anderson & B. Hunter, (Eds), *Computer literacy*. New York: Academic Press, 1982.
- Sussman, G. J., *A computer model of skill acquisition*. New York: American Elsevier, 1975.
- Talyzina, N., *The psychology of learning*. Moskou: Progress Publishers, 1981.
- Thibadeau, R., A. Just & P. A. Carpenter, *A model of the time course and content of reading*. *Cognitive Science*, 1982, 6, 157-203.
- Thorndike, E. L., *Mental discipline in highschool studies*. *Journal of Educational Psychology*, 1924, 15, 1-22; 83-98.
- Weir, S., *Logo and the exceptional child*. *Microcomputing* 1981, V, 48-52.
- Woodworth, R. S. & H. Schlosberg, *Experimental Psychology* (2nd ed). New York: Holt, Rhinehart & Winston, 1954.
- Zak, A., *Een experimenteel onderzoek naar reflectie bij jongere basisschoolleerlingen*. *Pedagogische Studiën*, 1980, 57, 358-365.

Curriculum vitae

H. Kunst (11-8-1945) studeerde van 1965 tot 1971 psychologie aan de Universiteit van Amsterdam. Was daarna tot 1975 op ZWO-basis werkzaam op het Instituut voor Cognitieforsching. Is sindsdien verbonden aan de vakgroep functieleer van de subfaculteit psychologie van de Rijksuniversiteit Utrecht. Onderzoeksinteressen zijn denken en probleemoplossen, vooral van jonge kinderen en leren programmeren.

Adres: Psychologisch Laboratorium, Rijksuniversiteit Utrecht, Varkenmarkt 2, 3511 BZ Utrecht.

Manuscript aanvaard 1-11-'83