

# Cognitieve effecten van het systematisch leren programmeren in LOGO: een onderwijsexperiment bij zesdeklassers

E. DE CORTE, L. VERSCHAFFEL en  
H. SCHROOTEN

Onderzoekscentrum voor Onderwijsleerprocessen, K.U. Leuven

## Samenvatting

*De hoofdbedoeling van deze studie was het ontwikkelen van een krachtige LOGO-leeromgeving gericht op het realiseren van de volgende voorwaarden voor het verwerven en de transfer van denkvaardigheden: 1. een voldoende kennis van de domeinspecifieke basisbegrippen van LOGO; 2. beheersing van de beoogde denkvaardigheden binnen de programmeercontext; en 3. deze denkvaardigheden ook in ten minste één ander inhoudsdomen kunnen toepassen. Een systematisch onderwijsexperiment werd opgezet in drie zesdeklassen volgens de voortoets-natoets opzet met controlegroep. In twee experimentele klassen vond LOGO-instructie plaats, met als voornaamste component het aanleren van een programmeerstrategie. In één van deze klassen werd de leerlingen bovendien uitdrukkelijk geleerd om de vaardigheden uit deze strategie te gebruiken in een ander vakgebied. In de controlegroep vond geen interventie plaats. Op het einde van het schooljaar werd vooreerst nagegaan of de vooropgestelde transfervoorwaarden vervuld waren. We stelden vast dat de eerste twee voorwaarden in beide experimentele groepen gerealiseerd waren; de resultaten met betrekking tot de derde voorwaarde waren minder positief. Verder werden verschillende transfertoetsen afgenomen. De analyse van de resultaten wees uit dat transfer werd bereikt in beide experimentele groepen; dit suggereert dat het vervullen van de eerste twee vermelde voorwaarden voldoende is om transfer van denkvaardigheden te bekomen.*

## 1 Theoretische achtergrond

Het verwerven van denk- en probleemoplossingsvaardigheden is een van de kerndoelstellingen van het basisonderwijs. Recent onderzoek heeft uitgewezen dat leren probleemoplossen neerkomt op het geïntegreerd verwerven en leren toepassen van drie soorten vaardigheden (De Corte & Verschaffel, 1987; ter perse; zie ook Perkins & Salomon, 1989), m.n.: 1. het flexibel kunnen hanteren van een goed georganiseerd domeinspecifiek kennisbestand, 2. de beheersing van heuristische methoden, en 3. het beschikken over metacognitieve kennis en vaardigheden. Daarenboven veronderstelt men vaak dat wie deze vaardigheden verworven heeft in een bepaald gebied, ze ook kan toepassen in andere contexten, m.a.w. er wordt verwacht dat *transfer* zal optreden.

De voorbije jaren is veelal beweerd dat leren programmeren een aangewezen hulpmiddel is voor het verwerven en de transfer van belangrijke heuristische en metacognitieve vaardigheden. Eén van de meest uitgesproken verdedigers van deze 'cognitieve effecten-hypothese' met betrekking tot leren programmeren is Papert. In zijn bekende boek *Mindstorms* (1980), waarvan in 1984 een Nederlandstalige vertaling verscheen, verdedigt hij de stelling dat leren programmeren in LOGO leidt tot de ontwikkeling van algemeen bruikbare denk- en probleemoplossingsvaardigheden, zoals plannen en 'debuggen', en tot het verwerven van krachtige begrippen, zoals variabele en recursie. Een ander kernidee uit Paperts visie is dat leren programmeren moet gebaseerd zijn op het principe van het zelfontdekkend leren: kinderen moeten hun kennis zelf actief opbouwen, door alleen en/of samen met andere leerlingen al spelend en experimenterend met de computer om te gaan; systematische sturing van het leerproces is daarbij uit den boze.

De resultaten van de pionierstudies over de cognitieve effecten van leren programmeren die in de eerste helft van de jaren tachtig ondernomen werden, boden evenwel weinig

steun voor deze optimistische hypothese. Een nadere analyse van deze studies suggereert echter dat de teleurstellende resultaten wellicht te wijten waren aan een aantal tekorten van de onderzoekingen zelf. Ten eerste hadden de leerlingen in deze studies onvoldoende programmeerkennis en -vaardigheden verworven, enerzijds door de korte duur van de instructie (gemiddeld ongeveer 25 tot 30 uren) en anderzijds door de afwezigheid van systematische begeleiding van het leerproces. Ten tweede ontbrak een expliciete gerichtheid op transfer (De Corte & Verschaffel, 1987; ter perse).

In 1985 startten we in het Leuven Onderzoekscentrum voor Onderwijsleerprocessen een project over de cognitieve effecten van leren programmeren in LOGO. De voornaamste doelstelling was het ontwerpen, implementeren en evalueren van een krachtige onderwijsleeromgeving, gericht op het verwerven en de transfer van algemene denkvaardigheden. Daarbij richtten we ons op een subset van de probleemoplossingsvaardigheden waarvan verwacht wordt dat deze beïnvloed worden door het leren programmeren: twee metacognitieve vaardigheden (plannen en debuggen), en twee heuristische (probleemdecompositie en het maken van een externe probleemrepresentatie).

Op basis van de voorafgaande studies over de effecten van leren programmeren en van de beschikbare literatuur over transfer in het algemeen, stelden we dat voor het bereiken van transfer via LOGO het vervullen van de volgende drie voorwaarden van belang is:

1. de leerlingen hebben voldoende domeinspecifieke kennis verworven (i.e. de primitieven en concepten van de LOGO-taal);
2. ze beheersen de beoogde heuristische en metacognitieve vaardigheden binnen de context van het programmeren in LOGO;
3. ze hebben uitdrukkelijk geleerd de denkvaardigheden ook in ten minste één ander inhoudsdomen toe te passen.

Omdat uit de voorgaande studies reeds voldoende gebleken was dat het realiseren van de eerste voorwaarde inderdaad een noodzakelijke, doch geen voldoende voorwaarde is voor het bekomen van transfer, hebben we ons toegespitst op het onderzoeken van de bijdrage van de laatste twee condities. De transferhypothese werd meer specifiek als volgt ge-

formuleerd: "Indien aan de eerste twee voorwaarden voldaan is, treedt er transfer van de beoogde denkvaardigheden op; aanvullende realisatie van de derde voorwaarde leidt tot een significant hoger transfereffect".

Het project werd uitgevoerd in twee fasen. Gedurende het schooljaar 1986-87 werd een *vooronderzoek* uitgevoerd, gericht op het ontwikkelen, implementeren en evalueren van een LOGO-onderwijsleeromgeving en op de constructie van toetsen over de beheersing van LOGO-kennis en -vaardigheden (transfervoorwaarden 1 en 2) (zie De Corte, Verschaffel, Hoedemaekers, Indemans & Schrooten, 1988). Om de transferhypothese te toetsen werd in de tweede fase (schooljaar 1987-88) een meer *systematisch onderwijsexperiment* uitgevoerd, waarin de geconstrueerde leeromgeving uitgebreid werd met een cruciale component, nl. het expliciet nastreven van transfer (transfervoorwaarde 3). In onderhavige bijdrage gaan we in op de opzet en de resultaten van deze tweede studie. Voor een meer uitvoerige beschrijving van dit onderzoek verwijzen we naar De Corte, Verschaffel, Schrooten, Indemans en Hoedemaekers (1990).

## 2 Proefgroep en onderzoeksofzet

### 2.1 Proefgroep

Het onderzoek vond plaats in drie zesdeklassen, elk bestaande uit 24 jongens van 11-12 jaar. De leerlingen hadden geen voorafgaande ervaring met computers in het algemeen, noch met LOGO in het bijzonder. Twee klassen fungeerden als experimentele groepen (E1 en E2); een derde klas vormde de controlegroep (C). In het begin van het schooljaar werden een intelligentietest en een schoolvorderingstoets afgenomen in de drie klassen. Er werden geen significante verschillen vastgesteld.

### 2.2 Opzet

Het toetsingsonderzoek werd uitgevoerd in de vorm van een voortoets-natoets opzet met controlegroep. In E1 werden de eerste twee voorwaarden voor transfer nagestreefd in een 60 uur durende LOGO-cursus, waarin de basisprimitieven en -concepten van de programmeertaal werden aangeleerd, evenals een strategie voor het systematisch schrijven van

LOGO-programma's. In het kader van deze programmeerstrategie, bestaande uit een planfase en een geïntegreerde uitvoerings- en controlefase, trainden we de beoogde denkvaardigheden (plannen, debuggen, probleemdecompositie, en constructie van een externe probleemrepresentatie). In E2 werd daarenboven de derde transfervoorwaarde nagestreefd: de jongens uit deze klas leerden de verworven denkvaardigheden ook nog toe passen in een andere context, nl. bij het oplossen van complexe rekenvraagstukken. In de controlegroep vond geen interventie plaats.

### 3 LOGO-leeromgeving

Met het oog op de verwezenlijking van de eerste twee voorwaarden voor het bereiken van transfer, werd in beide experimentele groepen gedurende het ganse schooljaar elke week een LOGO-namiddag georganiseerd. De instructie werd gegeven door het onderzoeksteam in samenwerking met de klasleerkracht. De computerklas, die direct toegankelijk was vanuit het gewone klaslokaal, was uitgerust met negen Philips MSX-computers. De LOGO-cursus bestond uit twee belangrijke componenten, nl. het aanleren van de LOGO-primitieven en -concepten (eerste transfervoorwaarde) en het trainen van een programmeerstrategie (tweede transfervoorwaarde).

#### 3.1 Aanleren van LOGO-primitieven en -concepten (eerste transfervoorwaarde)

De inhoud van de LOGO-cursus werd beperkt tot de zogenaamde 'turtle-graphics', of het tekenen met de schildpad. Bovendien werden enkel deze primitieven aangebracht, die we noodzakelijk achtten om de beoogde programmeerstrategie met succes te kunnen aanleren. Meer bepaald kwamen de volgende LOGO-noties aan de orde: maken van eenvoudige tekeningen met de elementaire primitieven (VT/AT, LI/RE, PO/PN), werken met het HERHAAL-commando, schrijven van procedures en superprocedures.

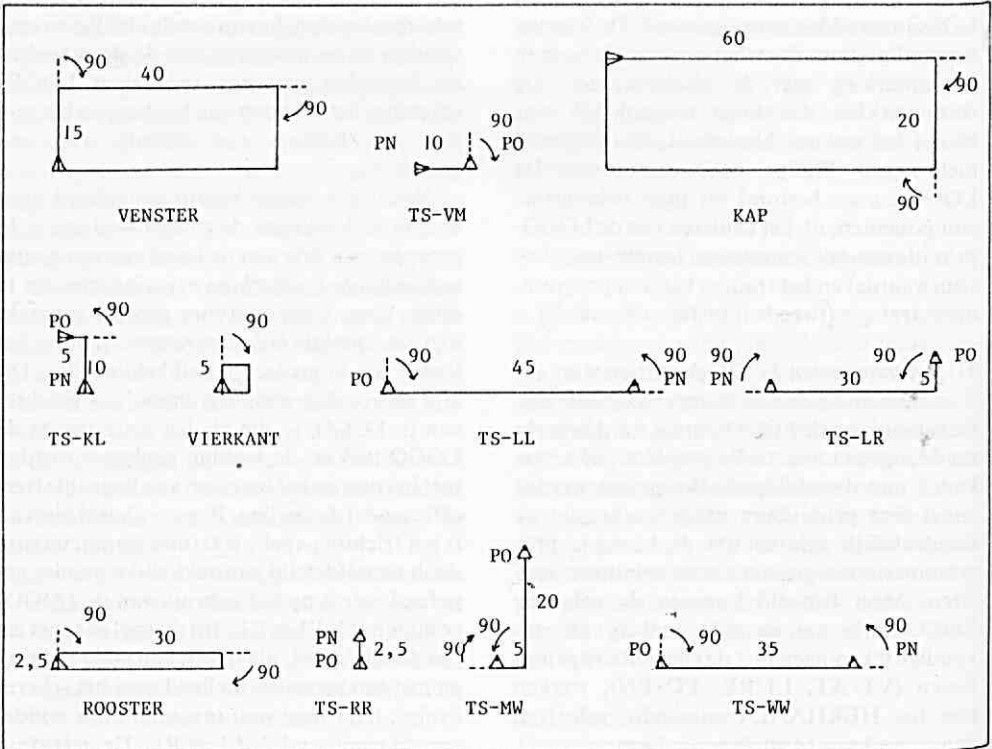
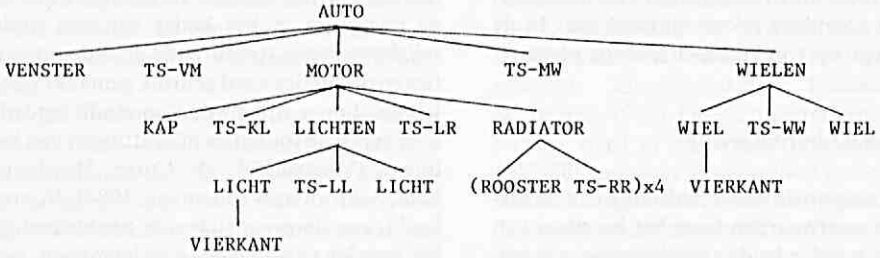
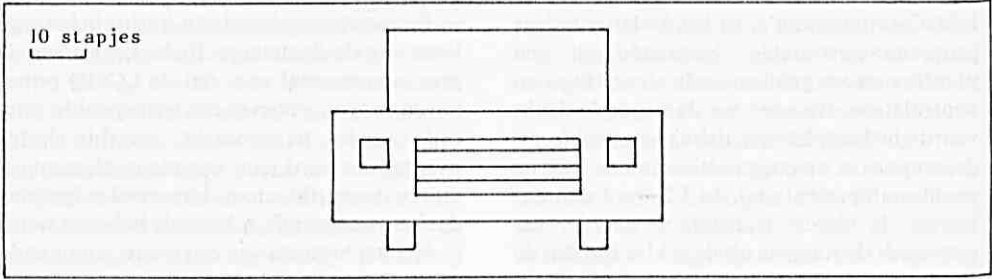
Uitgaande van een gematigd constructivistische visie op leren (De Corte, 1990), gingen we ervan uit dat een krachtige onderwijsleeromgeving gekenmerkt wordt door een evenwicht tussen zelfontdekkend leren enerzijds, en meer systematische instructie en sturing anderzijds.

Een eerste voorbeeld van sturing in het aanleren van de domeinspecifieke kennis van de programmeertaal was, dat de LOGO-primitieven en -concepten in een welbepaalde volgorde werden aangebracht, waarbij slechts overgegaan werd naar een nieuw element als uit een tussentijdse toets kon worden besloten dat het voorgaande voldoende beheerst werd.

Het aanbrengen van een nieuw commando of begrip geschiedde bovendien op een systematische wijze, meestal via *demonstratie* aan de computer in het kader van een *onderwijsleergesprek*. Gedurende deze demonstraties en discussies werd gebruik gemaakt van de kennis die we tijdens de voorstudie opdeden over typische fouten en misvattingen van leerlingen (Verschaffel, De Corte, Hoedemakers, Schrooten & Indemans, 1987). Bijvoorbeeld, om de geconstateerde problemen met het bepalen van hoeken te ondervangen, werd uitvoerig aandacht besteed aan: het zelf schildpad spelen, het op een duidelijke en eenvoudige wijze aangeven van de draaihoeken op het plan van een te maken LOGO-tekening, het schatten van hoeken, en het meten van hoeken met behulp van een graadboog.

Nadat een nieuw begrip aangeleerd was, kregen de leerlingen de gelegenheid om er in groepjes van drie aan de hand van een aantal *welomlijnde opdrachten* ervaring mee op te doen. Soms werd daarvoor gebruik gemaakt van z.g. speciale *hulpprogramma's*, die in het kader van dit project ontwikkeld werden. Dit zijn eenvoudige spelprogramma's – geschreven in LOGO – die als het ware tussen de LOGO-taal en de leerling geplaatst worden met het oog op het bereiken van bepaalde leereffecten bij de leerling. Een voorbeeld hiervan is het 'richting-spel', waarmee op een intense doch terzelfdertijd aantrekkelijke manier geoefend wordt op het gebruik van de LOGO-commando's LI en RE. Bij dit spel gaat het erom de schildpad, die in een bepaalde richting en met een bepaalde snelheid over het scherm kruipt, naar haar nest te sturen door middel van de commando's LI en RE. De oefenfase werd afgesloten met een leergesprek, waarin ervaringen, moeilijkheden, vondsten, strategieën, enz. van de leerlingen samengebracht en besproken werden.

Met het oog op kennisintegratie kregen de leerlingen telkens de gelegenheid om een nieuw verworven LOGO-begrip samen met



Figuur 1 Plan voor het schrijven van een LOGO-programma voor de tekening van de auto.

het eerder geleerde toe te passen in een meer omvangrijk *zelfgekozen* project.

### 3.2 *Aanleren van een strategie voor het schrijven van LOGO-programma's (tweede transfervoorwaarde)*

De aangeleerde strategie voor het schrijven van een LOGO-programma omvat twee fasen: 1. een planfase, die los van de computer plaatsvindt, en 2. een geïntegreerde uitvoerings- en controlefase aan de computer.

In de *planfase* onderscheiden we de volgende drie stappen (zie Figuur 1):

- Maken van een tekening van wat er precies op het scherm moet verschijnen.
- Opstellen van een boomdiagram waarin de complexe tekening opgedeeld wordt in delen, totdat men louter elementaire bouwstenen bekomt waarvan de programmering in LOGO als 'kinderspel' kan beschouwd worden (bijv. vierkant). Dit diagram duidt meteen de volgorde aan waarin de diverse onderdelen op het scherm getekend moeten worden. In het boomdiagram worden tevens de 'tussenstukken'<sup>1</sup> vermeld die de samenstellende delen verbinden.
- Apart tekenen van elke component, waarbij voor elk deel en elk tussenstuk de grootte van de afstanden en de draaihoeken worden aangegeven, evenals de begin- en eindpositie.

In deze eerste fase komen de volgende denkvaardigheden aan bod: plannen, decompositie (opsplitsen van een complexe figuur in delen) en het maken van een externe representatie (tekenen van een boomdiagram). De vierde vaardigheid die we beoogden te trainen, nl. het debuggen, vinden we terug in de tweede fase.

Pas nadat al de genoemde voorbereidingsactiviteiten beëindigd zijn, gaat de leerling naar de computer om het programma in te tikken. In de *uitvoerings- en controlefase* van de programmeerstrategie worden de volgende twee nauw met elkaar verbonden principes gehanteerd: 1) neerwaarts programmeren en 2) geïntegreerd uitvoeren en controleren.

- *Neerwaarts programmeren*. In plaats van een opwaartse aanpak, waarbij eerst alle enkelvoudige procedures van het laagste niveau van het boomdiagram geschreven worden en pas daarna overgegaan wordt tot het definiëren van de hoger gelegen pro-

cedures, hebben wij gekozen voor een neerwaartse aanpak. Dit houdt in dat begonnen wordt met het schrijven van de procedure uit het hoogste niveau van het boomdiagram; dit noemen we de 'moederprocedure'. Daarna wordt achtereenvolgens elke component uit de moederprocedure nader gespecificeerd, totdat het laagste niveau van het boomdiagram bereikt wordt.

- *Regelmatig controleren en debuggen*. In plaats van eerst de moederprocedure en alle deelprocedures in te tikken en pas daarna over te gaan tot het controleren en zondig verbeteren van alle invoer, hebben wij voor een geïntegreerde uitvoerings- en controleprocedure gekozen. Dit houdt in dat elke nieuwe procedure onmiddellijk getest wordt via het intikken van de naam van de moederprocedure. Het doel daarvan is tweeledig: 1) op het scherm verschijnt het resultaat van de tot dan toe ingevoerde procedures, wat de gelegenheid biedt om te controleren of de tekening en de schildpadpositie tot dan toe correct zijn; en 2) in de foutmelding ("ik ken geen...") staat vermeld welke procedure vervolgens geschreven moet worden.

Wat deze principes concreet inhouden, illustreren we weerom aan de hand van de taak uit Figuur 1. Overeenkomstig het principe van neerwaarts programmeren, tikt de leerling eerst de moederprocedure in:

LEER AUTO<sup>2</sup>

VENSTER TS-VM MOTOR TS-MW  
WIELEN EIND

Zoals het tweede principe voorschrijft, wordt vervolgens de moederprocedure (AUTO) uitgevoerd. Nadat de leerling het commando AUTO heeft ingetikt, verschijnt op het scherm de mededeling: "ik ken geen VENSTER in AUTO". Daarop definieert de leerling de eerste procedure uit de moederprocedure:

LEER VENSTER

HERHAAL 2 [VT 15 RE 90 VT 40 RE  
90] EIND

Wanneer vervolgens weerom de naam van de moederprocedure (AUTO) ingetikt wordt, tekent de schildpad het venster. De foutmelding geeft aan dat de procedure voor de verbinding tussen het venster en de motor (TS-VM) de volgende procedure is die geleerd moet worden. De leerling tikt in:

## LEER TS-VM

### PO RE 90 AT 10 PN EIND

Bij de daaropvolgende toetsing (AUTO) wordt het venster getekend en gaat de schildpad daarna naar de beginpositie voor de motor. Dan schrijft de leerling de procedure voor de motor. Bijvoorbeeld:

## LEER MOTOR

### KAP TS-KL LAMPEN TS-LR RADIA-TOR EIND

Als daarna de moederprocedure (AUTO) wordt uitgevoerd, tekent de schildpad opnieuw het venster en het verbindingstuk tussen het venster en de motor, en geeft vervolgens de boodschap "ik ken geen KAP in MOTOR". De volgende procedure die geleerd moet worden is de procedure KAP.

Dit procédé wordt herhaald tot alle delen en tussenstukken van MOTOR en vervolgens alle bestanddelen van de overige componenten uit de moederprocedure gedefinieerd zijn.

Het expliciet aanleren van deze programmeerstrategie was de belangrijkste component van de LOGO-cursus. Het instructiesysteem werd reeds een eerste keer uitgeprobeerd in het vooronderzoek (De Corte, e.a., 1988). Dit leidde tot de volgende bevindingen. Een groot deel van de leerlingen maakte niet spontaan een boomdiagram wanneer ze voor een complexe figuur een programma moesten schrijven. Dit kwam omdat zij het nut van de aangeleerde strategie niet inzagen. Maar ook wanneer we hen daar expliciet om vroegen, waren verschillende leerlingen niet in staat een correct boomdiagram te maken. Rekening houdend met deze ervaringen en bevindingen, werd de leeromgeving op enkele punten bijgestuurd: in het eigenlijke onderzoek werd de beoogde strategie op een meer systematische wijze aangebracht, gebruik makend van een aantal technieken die in de recente literatuur over het aanleren van denkvaardigheden naar voren komen als zeer belangrijk, zoals 'modelling', 'scaffolding', 'coaching', 'fading', 'articulation', 'reflection' and 'exploration' (Collins, Brown & Newman, in press). Daarnaast trachtten we de leerlingen meer te motiveren om de aangeleerde werkwijze te gebruiken.

Eerst werd de strategie in zijn geheel door een 'expert' (een lid van het onderzoeksteam) gedemonstreerd en werden de verschillende componenten geëxpliciteerd. Daardoor kon-

den de leerlingen een mentaal model van de noodzakelijke stappen en processen ontwikkelen, dat ze bij het zelfstandig toepassen voor ogen konden houden ('*modelling*').

Daarna werd elke component van de strategie gedetailleerd behandeld en afzonderlijk ingeoeft; zo nodig werd directe hulp geboden bij het uitvoeren van de taak ('*scaffolding*'). Dit hield o.m. in dat de leerkracht in een eerste fase die aspecten van de strategie voor zijn rekening nam die de leerlingen nog niet beheersten. Ook werd in de beginfase van het inoefenen van de planstrategie een computerprogramma gebruikt, dat hulp biedt bij het opstellen van een boomdiagram. Dit programma, dat in het kader van onderhavig onderzoek ontwikkeld werd, stuurt het proces van het opsplitsen van een tekening in delen door gerichte vragen te stellen (bijv. Is MOTOR eenvoudig genoeg? Uit welke delen bestaat MOTOR?). Via het meermaals gebruiken van dit programma voor het maken van boomdiagrammen voor verschillende tekeningen, werd ernaar gestreefd dat de leerlingen de werkwijze voor het plannen zouden internaliseren, om ze daarna zelfstandig te kunnen toepassen.

Vervolgens oefenden de leerlingen in groepjes van twee à drie de volledige programmeerstrategie in aan de hand van steeds moeilijkere tekeningen ('*exploration*'). Daarbij konden zij elk moment teruggrijpen naar kaarten waarop de voornaamste stappen van de strategie vermeld stonden, tesamen met nuttige tips om de onderscheiden fasen goed door te komen. Bij de uitvoerings- en controlefase aan de computer namen de leerlingen afwisselend de rol van 'uitvoerder' en van 'controleur' waar. Aanvankelijk werden de groepjes intensief begeleid ('*coaching*'), o.a. door het geven van aanwijzingen en terugkoppeling; doch afhankelijk van de individuele mogelijkheden en behoeften van de leerlingen verminderde deze hulp van de begeleiders gaandeweg ('*fading*'). Bij het geven van hulp werd expliciet rekening gehouden met de mogelijke fouten, misvattingen en moeilijkheden die uit het vooronderzoek gebleken waren. Zo werden bijvoorbeeld de beoordelingscriteria en de foutenclassificatie voor boomdiagrammen die afgeleid werden uit het vooronderzoek, aan de leerlingen kenbaar gemaakt en frequent gehanteerd – zowel door de begeleiders als door de leerlingen zelf – bij de

bespreking van de boomdiagrammen die gemaakt werden in het kader van de verschillende projecten in de loop van het schooljaar.

Gedurende de hele instructie en training van de strategie werden de leerlingen gestimuleerd om de procedures en technieken die ze aanwendden te verwoorden ('*articulation*'), en om deze te vergelijken met die van andere kinderen en met het expertmodel ('*reflection*'). Een belangrijke instructietechniek met het oog op het bereiken van transfer, was het expliciet en intentioneel *abstraheren* van de getrainde denkvaardigheden.

Om de leerlingen sterker te motiveren tot het gebruiken van de aangeleerde strategie werd ruime aandacht besteed aan het expliciet aantonen van het nut en de waarde ervan. Zo werden oefeningen gemaakt waarbij de leerlingen programma's moesten schrijven voor lichte variaties op eenzelfde tekening, waarbij de voordelen van het planmatig en modulair werken tot uiting kwamen. Ook werd er gedemonstreerd dat het eigenlijke programmeerwerk aan de computer en het verbeteren van fouten veel gemakkelijker en efficiënter verloopt wanneer uitgegaan wordt van een volledig en correct plan.

Om te vermijden dat de leerlingen een aversie zouden ontwikkelen voor een te rigide en strak opgelegde strategie, lieten we toe dat ze de aangeleerde werkwijze minder nauwgezet volgden naarmate het schooljaar vorderde. Zo moesten bijvoorbeeld in het begin alle tussentukken beginnen met PO en eindigen met PN; naarmate de leerlingen onderscheid leerden maken tussen situaties waar dit noodzakelijk is en waar niet, mocht dit achterwege gelaten worden. Dit is perfect in overeenstemming met onze opvatting over het leerproces. Immers, we beschouwen de volledige procedure als een geschikte aanvangshandlungsstructuur, waarin een adequaat oplossingsproces zeer uitvoerig veruitwendigd wordt. Doch in de loop van het leerproces kunnen allerhande verkortingen en verinnerlijkingen plaatsvinden.

#### 4 *Expliciet nastreven van transfer (derde transfervoorwaarde)*

In de tweede experimentele groep werden op het einde van de LOGO-cursus een vijftal lessen ingelast, waarbij de leerlingen de algeme-

ne denkvaardigheden onderwezen in de LOGO-context (nl. plannen, debuggen, probleemdecompositie en constructie van een externe representatie) ook leerden toepassen bij het oplossen van complexe rekenvraagstukken. Meer bepaald leerden ze een strategie bestaande uit twee fasen: een planfase en een uitvoerings- en controlefase. Zoals in LOGO, bestaat het plannen van een oplossingsweg voor een complex vraagstuk erin het probleem op te splitsen in deelproblemen (decompositie-heuristiek); het resultaat van deze planactiviteiten kan eveneens weergegeven worden in een boomdiagram (maken van een externe representatie). In de uitvoerings- en controlefase wordt successief ieder deelprobleem opgelost en worden de antwoorden onmiddellijk gecontroleerd.

## 5 *Meetinstrumenten*

### 5.1 *Criteriumtoetsen*

Vooraleer over te gaan tot de toetsing van het transfereffect werd onderzocht of aan de transfervoorwaarden voldaan was. In beide experimentele klassen werden één maand voor het einde van de leergang verschillende criteriumtoetsen afgenomen in verband met de eerste twee transfervoorwaarden. Voor elke toets werd per klas onderzocht of meer dan 75% van de leerlingen het vooropgestelde individueel criterium (nl. 2/3 van het maximum aantal punten) behaalde. Wanneer dit niet zo was, beschouwden we de betreffende voorwaarde als niet voldoende vervuld. In dit laatste geval volgde hierop een remediëring en hertoetsing voor die leerlingen die het criterium niet bereikten. De toetsen in verband met vraagstukken (derde transfervoorwaarde) werden slechts één keer afgenomen, daar er geen tijd beschikbaar was voor remediëring en hertoetsing; dit gebeurde vanzelfsprekend alleen in E2.

De toets over de kennis van de LOGO-primitieven en -concepten (transfervoorwaarde 1) bestond uit twee reeksen van opgaven: bij de eerste reeks moesten de leerlingen aangeven welk van de vier alternatieve LOGO commando's of procedures een gegeven schermffect zou produceren; bij de tweede reeks was de opdracht de tekening aan te duiden die door een gegeven programma op het scherm gemaakt zou worden.

Daarnaast waren er drie toetsen om na te gaan of de leerlingen de programmeerstrategie verworven hadden (transfervoorwaarde 2). Twee ervan hadden betrekking op de planfase. In de eerste toets bestond elke opgave uit een tekening met twee boomdiagrammen, waarvan slechts één correct was; gevraagd werd het juiste boomdiagram voor de betreffende tekening aan te duiden. In de tweede toets over de planfase moesten de leerlingen zelfstandig een adequaat boomdiagram construeren voor een gegeven tekening. De derde toets betrof de uitvoerings- en controlefase. In tegenstelling tot de twee voorgaande toetsen, werd deze toets individueel aan de computer afgenomen. De opdracht was een LOGO-programma te schrijven voor een relatief complexe tekening, uitgaande van een gegeven plan. Om het debuggingsproces te kunnen analyseren, werden er in dit plan vier fouten verwerkt.

Om de beheersing van de strategie voor het oplossen van samengestelde rekenvraagstukken (transfervoorwaarde 3) na te gaan, werden analoge toetsen gebruikt. Zoals bij LOGO hadden twee toetsen betrekking op de planfase: in de eerste moesten de leerlingen het beste van twee gegeven boomdiagrammen aanduiden voor een reeks van vraagstukken; in de tweede toets werden ze gevraagd zelfstandig een adequaat boomdiagram te construeren voor een reeks gegeven vraagstukken. De derde toets betrof de uitvoerings- en controlefase; de opdracht was een aantal vraagstukken op te lossen, uitgaande van gegeven boomdiagrammen.

## 5.2 *Transfertoetsen*

De leerlingen uit de drie proefgroepen kregen een aantal toetsen aangeboden om na te gaan in welke mate ze in staat waren de in E1 en E2 aangeleerde denkvaardigheden spontaan en zelfstandig toe te passen in nieuwe probleem-situaties.

De doolhof-toets, een computertoets voor planvaardigheid, werd individueel aan de machine afgenomen. De leerling kreeg achtereenvolgens drie opgaven aangeboden, waarbij een hongerige kever via de kortste weg naar drie slaatjes gestuurd moest worden. De doelhoven waren zo geconstrueerd, dat plannen noodzakelijk was om een goed resultaat te bekomen.

In de foutendetectie-toets werd zowel het

plannen als debuggen gemeten. In drie opgaven werd telkens een dagelijkse situatie geschetst, waarbij persoon X aan persoon Y een reeks instructies gaf om een bepaalde taak te volbrengen (McCoy Carver, 1986). Persoon Y voerde deze instructies nauwgezet uit, maar het verkregen resultaat kwam niet overeen met het gewenste, doordat persoon X een verkeerde instructie gegeven had. De opdracht was om deze foutieve instructie op te sporen en de instructielijst zodanig te veranderen, dat het gewenste resultaat wel bereikt zou worden.

In de blokken-toets bestond de opdracht erin een reeks gegeven complexe figuren te beschrijven in termen van hun samenstellende delen, daarbij gebruik makend van een reeks gegeven elementen. Deze activiteit vereiste de toepassing van de decompositie-heuristiek.

De weekdagen-toets omvatte een reeks opgaven, die alle een variant waren op de volgende vraag: "Welke dag was het eergisteren, als het morgen zaterdag is?" (Bransford & Stein, 1984). Bij het oplossen van deze taken was het maken van een probleemdecompositie en een externe probleemrepresentatie noodzakelijk.

Het gebruik van externe representaties werd ook nagegaan in de dozen-toets, die bestond uit een aantal items, zoals "In een grote doos zitten twee middelgrote dozen. In de eerste middelgrote doos zitten drie kleine doosjes. In de tweede middelgrote doos zitten vier kleine doosjes. Hoeveel dozen zijn er in totaal?" (Bransford & Stein, 1984).

## 6 *Resultaten*

### 6.1 *Resultaten van de criteriumtoetsen*

Tabel 1 vermeldt voor elke criteriumtoets het aantal leerlingen in E1 en E2 dat het vooropgestelde beheersingscriterium (2/3 van de maximumscore) bereikte op het einde van het schooljaar.

Uit de resultaten op de toets over LOGO-kennis bleek dat op het einde van de leergang de meeste leerlingen uit beide experimentele klassen (21 en 17 in E1 resp. E2) een voldoende niveau van kennis van de LOGO-primitieven en concepten bereikt hadden (transfervoorwaarde 1).

De bevindingen op de toetsen betreffende de programmeerstrategie (transfervoorwaarde 2) waren eveneens erg positief. De resulta-



Tabel 1 Aantal leerlingen in E1 en E2 dat het beheersingscriterium bereikte op de toetsen over de drie transfer-voorwaarden

Transfervoorwaarden	Criteriumtoetsen	Resultaten	
		E1 (n = 24)	E2 (n = 24)
1. LOGO-kennis	LOGO-kennis	21	17
2. Denkvaardigheden bij LOGO	Plannen 1	19	23
	Plannen 2	20	19
	Uitvoeren/controleren	20	22
3. Denkvaardigheden bij vraagstukken	Plannen 1	-	22
	Plannen 2	-	10
	Uitvoeren/controleren	-	22

ten van de eerste toets over de planfase toonden aan dat in E1 en E2 resp. 19 en 23 leerlingen voldoende in staat waren een reeks gegeven boomdiagrammen op hun juistheid te beoordelen. Uit de tweede toets over de planfase bleek dat in E1 en E2 resp. 20 en 19 leerlingen erin slaagden zelfstandig een adequaat boomdiagram te construeren voor een gegeven tekening. Bij de toets over de uitvoerings- en controlefase stelden we vast dat de aangeleerde werkwijze door de meeste leerlingen (20 en 22) nauwgezet gevolgd werd: bij de opbouw van hun programma gingen ze neerwaarts tewerk; de ingevoerde procedures werden regelmatig getest door het uitvoeren van de moederprocedure, en wanneer een verkeerd schermefect optrad, verliep het opzoeken en verbeteren van fouten op een systematische manier. We kunnen dus stellen dat de beoogde denkvaardigheden, nl. plannen, debuggen, probleemdecompositie, en gebruik van een externe representatie, binnen de LOGO-context voldoende beheerst werden. Samenvattend kunnen we besluiten dat op het einde van de LOGO-cursus in beide experimentele klassen de eerste twee transfervoorwaarden vervuld waren.

Uit Tabel 1 blijkt verder dat de resultaten van de criteriumtoetsen in verband met de derde transfervoorwaarde, m.n. het kunnen toepassen van de geleerde vaardigheden bij het oplossen van andersoortige problemen, minder overtuigend waren. Immers, juist op de meest cruciale toets over deze voorwaarde, nl. deze waarbij de leerlingen zelfstandig de geleerde werkwijze voor het plannen moesten

toepassen bij het oplossen van een reeks complexe vraagstukken (tweede toets), behaalden slechts 10 van de 24 leerlingen uit de tweede experimentele klas het criterium. Er was evenwel geen tijd meer beschikbaar voor remediëring, omdat het schooljaar op zijn einde liep. Omwille van deze laatste bevinding, was het meteen uitgesloten om op basis van de gegevens van dit onderzoek een beslissende toetsing te doen van het tweede gedeelte van de onderzoekshypothese, namelijk dat bij de verwezenlijking van de derde voorwaarde het transfereffect zal verhogen.

## 6.2 Resultaten van de transfertoetsen

Om de transfer van elk van de vier denkvaardigheden na te gaan werd een multivariate variantie-analyse (MANOVA) toegepast op de gegevens van de transfertoetsen. Afzonderlijke analyses werden uitgevoerd voor de vergelijking van E1 versus C enerzijds, en E1 versus E2 anderzijds. Volgens onze oorspronkelijke hypothese, zou E1 beter presenteren dan C, en zou E2 bovendien E1 overtreffen.

De multivariate analyse toonde een significant verschil tussen E1 en C op de maten voor debuggen, probleemdecompositie en externe representatie in het voordeel van E1 (zie Tabel 2). Hieruit kunnen we concluderen dat de LOGO-cursus een transfereffect teweeg bracht voor drie van de vier getrainde vaardigheden bij deze leerlingen bij wie de eerste twee transfervoorwaarden voldaan waren.

Voor de planvaardigheid werd geen significant verschil gevonden: beide groepen presteerden even goed op de toetsen die op deze

Tabel 2 *Vergelijking van E1 vs. C op de transfertoetsen (MANOVA)*

Denkvaardigheden	Transfertoetsen	Resultaten
Plannen	Doolhof, Foutendetectie	.249
Probleemdecompositie	Blokken, Weekdagen	< .001
Externe representatie	Weekdagen, Dozen	< .001
Debuggen	Foutendetectie	.007

vaardigheid beroep deden. Dit is enigszins verwonderlijk, daar de plancomponent de kern van de aangeleerde strategie uitmaakte. De meest voor de hand liggende verklaring hiervoor is naar onze mening dat de toets voor het meten van de planvaardigheid slechts een gebrekkige validiteit bezat.

Zoals verwacht kon worden op basis van het niet vervuld zijn van de derde transfervoorwaarde, maakte E2 niet meer vooruitgang dan E1.

## 7 *Samenvatting en discussie*

In onderhavige bijdrage werden de opzet en de resultaten beschreven van een onderwijsexperiment waarin we een LOGO-leeromgeving implementeerden, gericht op het verwerven en de transfer van algemeen bruikbare denkvaardigheden. Het onderzoek werd uitgevoerd in drie zesdeklassen, volgens een voortoetsnatoets opzet met controlegroep.

Op basis van de recente literatuur formuleerden we drie voorwaarden die van belang zijn voor het bereiken van transfer van denkvaardigheden. De eerste twee voorwaarden – het verwerven van een voldoende niveau van domeinspecifieke kennis enerzijds, en de beheersing van de beoogde denkvaardigheden binnen de programmeercontext anderzijds – trachtten we in beide experimentele klassen te realiseren in een LOGO-cursus. In de tweede experimentele klas werd daarenboven expliciet transfer nagestreefd (derde voorwaarde) in een bijkomende vraagstukcursus.

Alvorens het transfereffect na te gaan, onderzochten we of de nagestreefde *transfervoorwaarden* vervuld waren. Uit de resultaten van de criteriumtoetsen over LOGO bleek dat op het einde van de leergang de eerste twee voorwaarden in beide experimentele klassen voldoende gerealiseerd waren. De meeste leerlingen hadden de aangeleerde primitieven en concepten van de LOGO-taal goed verworven

en waren in staat de geleerde denkvaardigheden toe te passen binnen de LOGO-context. Dat deze gunstige uitkomsten werden bereikt, kan onzes inziens toegeschreven worden aan de zorgvuldig ontwikkelde onderwijsleeromgeving, waarin tegemoet gekomen werd aan de tekortkomingen van de vroegere studies. Twee belangrijke kenmerken van deze leeromgevingen waren: 1) de relatief lange duur van de LOGO-instructie en 2) het evenwicht tussen zelfontdekkend leren en systematische instructie (bijv. het expliciet aanleren van een programmeerstrategie). Toch willen we waarschuwen voor een al te groot optimisme omtrent de veralgemeenbaarheid van deze resultaten. Men mag namelijk niet uit het oog verliezen dat de gehanteerde leeromgeving allesbehalve representatief is voor de feitelijke situatie in de meeste lagere scholen anno 1989: er werd gedurende 60 uur LOGO-instructie gegeven door drie ervaren begeleiders in samenwerking met de leerkracht en daarbij waren negen computers beschikbaar. En zelfs dan nog bleven bij verscheidene leerlingen lacunes en hardnekkige misvattingen bestaan, die slechts via geïndividualiseerde remedieering verholpen konden worden.

We slaagden er niet in de derde transfervoorwaarde te realiseren. Zelfs na een periode van expliciete instructie in het aanwenden van de in de LOGO-context geleerde probleemoplossingsvaardigheden bij het oplossen van complexe rekenvraagstukken, bleken de meeste leerlingen in E2 onvoldoende in staat deze vaardigheden zelfstandig toe te passen bij de vraagstukken uit de betreffende criteriumtoets. Deze teleurstellende resultaten zijn waarschijnlijk toe te schrijven aan de stevig verankerde, maar eerder ongunstige aanpakwijzen en denkbeelden in verband met wiskundige problemen in het algemeen en met vraagstukken in het bijzonder, die bij deze leerlingen vooraf reeds bestonden ten gevolge van jarenlang vraagstukkenonderwijs (Sowder, 1988). Allicht volstonden de duur en de

aard van de instructie niet om wezenlijke veranderingen op strategisch vlak te bewerkstelligen.

Wat het eigenlijke *transfereffect* betreft, werden bemoedigende resultaten verkregen. Bij de leerlingen die de LOGO-cursus volgden (E1), werd voor drie van de vier gemeten vaardigheden, nl. debugging, decompositie, en externe representatie, transfer bereikt. Inmiddels zijn in andere recente studies gelijkaardige positieve resultaten geboekt (zie o.m. Mayer, 1988; voor een overzicht, zie De Corte & Verschaffel, 1987, ter perse). Een gemeenschappelijke factor in deze succesrijke studies is de expliciete instructie van de beoogde denkvaardigheden in de LOGO-context, waarbij deze vaardigheden bewust en intentioneel geabstraheerd worden; Salomon en Perkins (1987) noemen dit 'mindful abstraction'.

Het tweede deel van onze verwachting, m.n. dat de expliciete instructie voor transfer het effect zou verhogen, werd niet ingelost: de leerlingen die bijkomend een vraagstukken-cursus gevolgd hadden (E2), behaalden op geen van de vier maten hogere resultaten dan deze uit E1. Toch zou het voorbarig zijn om hieruit af te leiden dat het expliciet nastreven van transfer naar verschillende domeinen geen gunstige invloed heeft op de algemene denkontwikkeling van leerlingen. Immers, in onderhavige studie kan het ontbreken van het verwachte transfereffect verklaard worden door het niet bereiken van de doelstelling van de vraagstukken-cursus, m.n. het leren toepassen van de denkvaardigheden verworven in de LOGO-context bij het oplossen van complexe rekenvraagstukken (= transfervoorwaarde 3). Daardoor kwam het toetsen van de hypothese in het kader van deze studie als het ware al bij voorbaat op de helling te staan. Mede op grond van recente literatuur (o.m. Gick & Holyoak, 1987; Perkins & Salomon, 1989), blijven we evenwel de hypothese aanhouden dat, indien men erin slaagt de derde voorwaarde te realiseren, er wel een verhoging van het transfereffect kan verwacht worden. Overigens werd in enkele meer recente studies (zie o.m. Littlefield, Delclos, Lever, Clayton, Bransford & Franks, 1988) wel steun voor deze hypothese verkregen.

Het uitgangspunt van onderhavig onderzoek was de vraag "Heeft het leren programmeren een invloed op de denkvaardigheden van ba-

sissschoolleerlingen?". Op basis van de verkregen onderzoeksresultaten en de opgedane ervaringen menen we hierop een gematigd positief antwoord te kunnen formuleren. *Ja*, de bevindingen suggereren dat leren programmeren in LOGO kan leiden tot het verwerven en de transfer van algemene denkvaardigheden. *Maar*, de positieve effecten vloeien allicht niet louter voort uit de unieke eigenschappen van LOGO als dusdanig, maar eerder uit de kwaliteiten van de onderwijsleeromgeving waarin de programmeertaal aangeleerd wordt. Met andere woorden, LOGO *op zich* biedt geen garantie voor het tot stand komen van denkvaardigheden, maar kan hiervoor een nuttig hulpmiddel zijn als het ingebed is in een krachtige leeromgeving, d.w.z. een situatie die aanleiding geeft tot het uitlokken bij de leerlingen van de vereiste leerprocessen om de beoogde onderwijsdoelen te bereiken.

Voortgezet onderzoek zou gericht moeten zijn op het meer precies identificeren en analyseren van de cruciale kenmerken van zulke leeromgevingen. Een sleutelvraag in dit verband is of, en in hoeverre de noodzakelijke aanvullende denkinstructie het best in de programmeeromgeving zelf wordt ingebouwd, en welk gedeelte beter in handen van de leerkracht kan blijven. Met andere woorden, moet er gewerkt worden in de richting van intelligente systemen die op basis van verfijnde diagnoses van de vaardigheden en aanpakstrategieën van de leerlingen aangepaste begeleiding kunnen voorzien? Of moeten deze activiteiten eerder uitgevoerd worden door de leerkracht en blijft de rol van de computer het best beperkt tot het bieden van een situatie die de leerlingen uitnodigt tot het maximaal aanwenden van hun eigen kennis en denkvermogens (Scardamalia, Bereiter, McLean, Swallow, & Woodruff, 1989)? Een voorbeeld van een systeem waarin de oriëntering naar algemene denkvaardigheden sterker door de computer zelf ondersteund wordt, is het Michael-systeem, dat ontwikkeld werd door de Afdeling Informatica van het Departement Computerwetenschappen van de K.U. Leuven, in samenwerking met onze onderzoeksgroep. Dit systeem stimuleert uitdrukkelijk een stapsgewijze opbouw en verfijning van programma's. Tevens biedt het verschillende mogelijkheden om de gebruiker te bevrijden van de syntactische aspecten van het programme-

ren, waardoor de aandacht kan toegespitst worden op de meer essentiële probleemoplossings- en ontwerpactiviteiten (Tollenaere, 1988).

Tot slot lijkt het ons aangewezen het verder onderzoeks- en ontwikkelingswerk niet te beperken tot programmeer-leromgevingen. Aangezien de vastgestelde cognitieve effecten allicht geen rechtstreeks resultaat zijn van het programmeren op zich, maar eerder van de gehele leeromgeving waarin dit ingebed wordt, kunnen we aan het leren programmeren geen monopoliepositie inzake denktraining toekennen. Bijgevolg mogen we veronderstellen dat, gegeven de investering van eenzelfde hoeveelheid tijd, inspanning en expertise, even efficiënte leeromgevingen kunnen gecreëerd worden in meer 'traditionele' inhoudsdomeinen; dit zou daarenboven de integratie van de denktraining in het reguliere curriculum ten goede komen. Hiermee willen we pleiten voor de aanpak van de denkscholing vanuit verschillende inhoudsdomeinen. Overigens vereist de totstandkoming van *algemeen bruikbare* vaardigheden dat de kinderen deze leren toepassen in verschillende vakgebieden en probleemsituaties. Dit kan enkel verwezenlijkt worden indien er vakoverschrijdend gepland wordt hoe men het transferegericht leren vorm zal geven.

## Noten

1. Bij de tussenstukken geldt de volgende afspraak. De naam van een tussenstuk bestaat uit twee delen: ts (van tussenstuk), gevolgd door de eerste letter van het vorige deel en de eerste letter van het volgende deel (bijv.: tussenstuk venster dak: ts-vd).
2. Om overzichtelijke programma's te bekomen raden we de kinderen aan om een nieuwe lijn te beginnen voor iedere procedure in een superprocedure. In deze tekst staan echter wegens plaatsgebrek alle procedures op één lijn.

## Literatuur

- Bransford, J. D. & B. S. Stein, *The ideal problem solver*. New York: Freeman, 1984.
- Collins, A., J. S. Brown & S. E. Newman, Cognitive apprenticeship: Teaching the craft of reading, writing and mathematics. In: L. B. Resnick (Ed.), *Cognition and instruction: Issues and agendas*. Hillsdale, NJ: Erlbaum, in press.
- Corte, E. De, Ontwerpen van krachtige onderwijsleeromgevingen. In: M. J. Ippel & J. J. Elshout (Eds.), *Training van hogere-orde denkprocessen*. Lisse: Swets & Zeitlinger, 1990.
- Corte, E. De & L. Verschaffel, Cognitieve effecten van leren programmeren. *Nederlands Tijdschrift voor de Psychologie*, 1987, 42, 364-372.
- Corte, E. De & L. Verschaffel, Leuvense perspectieven: Informatica en leren probleemoplossend denken in het onderwijs. *Onze Alma Mater*, ter perse.
- Corte, E. De, L. Verschaffel, E. Hoedemaekers, R. Indemans & H. Schrooten, *Computers en Leren Denken. Activiteitenverslag over het werkjaar 1987*. (Project "Computers en Leren Denken", Rapport nr. 6). Leuven: Onderzoekscentrum voor Onderwijsleerprocessen, K.U. Leuven, 1988.
- Corte, E. De, L. Verschaffel, H. Schrooten, R. Indemans & E. Hoedemaekers, *LOGO als springplank voor het verwerven van denkvaardigheden bij zesdeklassers* (Leuvense Pedagogische Bijdragen). Leuven: Acco, 1990.
- Gick, M. L. & K. J. Holyoak, The cognitive basis of knowledge transfer. In: S. M. Cormier & J. D. Hagman (Eds.), *Transfer of learning. Contemporary research and applications*. (Educational Technology Series). San Diego: Academic Press, 1987.
- Littlefield, J., V. R. Delclos, S. Lever, K. N. Clayton, J. D. Bransford & J. J. Franks, Learning LOGO: Method of teaching, transfer of general skills, and attitude toward school and computers. In: R. E. Mayer (Ed.), *Teaching and learning computer programming. Multiple research perspectives* (pp. 111-135). Hillsdale, NJ: Erlbaum, 1988.
- Mayer, R. E. (Ed.), *Teaching and learning computer programming. Multiple research perspectives*. Hillsdale, NJ: Erlbaum, 1988.
- McCoy Carver, S., *Transfer of LOGO debugging skill: Analysis, instruction and assessment*. (Doctoral dissertation). Pittsburgh, PA: Department of Psychology, Carnegie-Mellon University, 1986.
- Papert, S., *Mindstorms. Children, computers, and powerful ideas*. New York: Basic Books, 1980.
- Papert, S., *Computers en kinderen*. Amsterdam: Bert Bakker, 1984.

- Perkins, D.N. & G. Salomon, Are cognitive skills context-bound? *Educational Researcher*, 1989, 18, no. 1, 16-25.
- Salomon, G. & D.N. Perkins, Transfer of cognitive skills for programming: When and how? *Journal of Educational Computing Research*, 1987, 3, 149-169.
- Scardamalia, M., C. Bereiter, R.S. McLean, J. Swallow & E. Woodruff, Computer-supported intentional learning environments, *Journal of Educational Computing Research*, 1989, 5, 51-68.
- Sowder, L., Children's solutions of story problems. *Journal of Mathematical Behavior*, 1988, 7, 227-238.
- Tollenaere, T., *Een didactische programmeeromgeving* (niet-gepubliceerde licentiaatsverhandeling). Leuven: K.U. Leuven, Departement Computerwetenschappen, 1988.
- Verschaffel, L., E. De Corte, E. Hoedemaekers, H. Schrooten & R. Indemans, *Beheersing van de LOGO-kennis bij zesdeklassers na een jaar LOGO-instructie*. (Project "Computers en Leren Denken", Rapport nr. 5). Leuven: Onderzoekscentrum voor Onderwijsleerprocessen, K.U. Leuven, 1987.

## Curricula vitae

E. De Corte is gewoon hoogleraar aan de Faculteit der Psychologie en Pedagogische Wetenschappen, K.U. Leuven.

L. Verschaffel is bevoegdverklaard navorser bij het Nationaal Fonds voor Wetenschappelijk Onderzoek en buitengewoon docent aan de Faculteit der Psychologie en Pedagogische Wetenschappen, K.U. Leuven.

H. Schrooten is wetenschappelijk medewerker in het onderzoeksproject "Computers en Leren Denken" van het Onderzoekscentrum voor Onderwijsleerprocessen, Faculteit der Psychologie en Pedagogische Wetenschappen, K.U. Leuven.

Adres: Onderzoekscentrum voor Onderwijsleerprocessen, Departement der Psychologie en Pedagogische Wetenschappen, K.U. Leuven, Vesaliusstraat 2, B-3000 Leuven, België.

Manuscript aanvaard 12-3-90

## Summary

Corte, E. De, L. Verschaffel & H. Schrooten. 'Cognitive effects of learning to program in LOGO: A one-year study with sixth graders.' *Pedagogische Studiën*, 1990, 67, 293-305.

This study aimed at the development of a powerful LOGO learning environment for achieving the following conditions for the acquisition and transfer of thinking skills: 1) sufficient knowledge of the domain specific LOGO primitives and concepts; 2) mastery of the thinking skills within the programming context; and 3) ability to apply these skills in at least one other content domain. A systematic experiment was carried out in three sixth grade classes according to the pretest-posttest design with control group. A LOGO course, involving the training of a systematic programming strategy, was taught in two experimental classes. In one of these classes the pupils also learned how to apply the acquired skills in another content domain. The control group was a non-treatment group. At the end of the school year condition fulfilment was tested. The findings showed that the first two transfer conditions were fulfilled in both experimental groups; the results with respect to the third condition were less positive. Furthermore, a series of transfer tests were administered. Data analysis revealed that transfer was obtained in both experimental groups. This implies that fulfilment of the first two conditions mentioned above is sufficient for realizing transfer.