# Stichting NIOC en de NIOC kennisbank

Stichting NIOC (www.nioc.nl) stelt zich conform zijn statuten tot doel: het realiseren van congressen over informatica onderwijs en voorts al hetgeen met een en ander rechtstreeks of zijdelings verband houdt of daartoe bevorderlijk kan zijn, alles in de ruimste zin des woords.

De stichting NIOC neemt de archivering van de resultaten van de congressen voor zijn rekening. De website www.nioc.nl ontsluit onder "Eerdere congressen" de gearchiveerde websites van eerdere congressen. De vele afzonderlijke congresbijdragen zijn opgenomen in een kennisbank die via dezelfde website onder "NIOC kennisbank" ontsloten wordt.

Op dit moment bevat de NIOC kennisbank alle bijdragen, incl. die van het laatste congres (NIOC2023, gehouden op donderdag 30 maart 2023 jl. en georganiseerd door NHL Stenden Hogeschool). Bij elkaar bijna 1500 bijdragen!

We roepen je op, na het lezen van het document dat door jou is gedownload, de auteur(s) feedback te geven. Dit kan door je te registreren als gebruiker van de NIOC kennisbank. Na registratie krijg je bericht hoe in te loggen op de NIOC kennisbank.

Het eerstvolgende NIOC vindt plaats op donderdag 27 maart 2025 in Zwolle en wordt dan georganiseerd door Hogeschool Windesheim. Kijk op www.nioc2025.nl voor meer informatie.

Wil je op de hoogte blijven van de ontwikkeling rond Stichting NIOC en de NIOC kennisbank, schrijf je dan in op de nieuwsbrief via

www.nioc.nl/nioc-kennisbank/aanmelden nieuwsbrief

Reacties over de NIOC kennisbank en de inhoud daarvan kun je richten aan de beheerder:

R. Smedinga kennisbank@nioc.nl.

Vermeld bij reacties jouw naam en telefoonnummer voor nader contact.

# Denkniveaus bij algoritmen

***Jacob Perrenet, TU/e Informatica***
***Jan Friso Groote, TU/e Informatica***
***Eric Kaasenbrood, TU/e Informatica***

**Samenvatting**
*Onderzoek naar informaticaonderwijs staat nog in de kinderschoenen. Het kan voortbouwen op de traditie bij het wiskundeonderwijs. Ons onderzoek, uitgevoerd bij de Bacheloropleiding aan de Technische Universiteit Eindhoven, vindt zijn inspiratie in theorieën over wiskundige denkniveaus. Vier abstractieniveaus werden verondersteld in het denken over algoritmen. Een vragenlijst werd geconstrueerd om bij verschillende groepen studenten het beantwoordingsniveau te meten. De resultaten toonden de aanwezigheid van niveauverschillen. Tussen opvolgende jaargroepen trad niveauverhoging op en er was groei in de loop van een jaar. Samenhang tussen gemeten denkniveau en behaald tentamencijfer bij algoritmevakken was er echter nauwelijks. Ook docenten vulden de vragenlijst in.*
**Keywords:** algoritmen, abstractie, curriculum, didactisch onderzoek

## Informaticaonderwijs en onderwijsonderzoek aan het begin

Franquinet (2004) schetst de gang van het informaticaonderwijs naar de officiële status in het vo-curriculum. Zo ver is het nog niet. Pas wanneer een vak een volwassen plaats heeft veroverd in primair of secundair onderwijs komt onderzoek naar het onderwijzen en leren van dat vak goed op gang. Ook wereldwijd is er nog lang geen traditie, zoals bij het wiskundeonderwijs, al was het maar, omdat informatica nog zo'n jonge wetenschap is. Net zoals de informatica door de wiskunde is beïnvloed, kan ook het onderzoek van het informaticaonderwijs zich laten inspireren door wat binnen de wereld van het wiskundeonderwijs reeds tot stand is gebracht (Almstrum, e.a., 2002).

## Niveautheorie

Bij niveautheorie denken we in Nederland in de eerste plaats aan het werk van Van Hiele (1986). Internationaal zo mogelijk nog bekender is het verwante werk van Skemp (zie bijvoorbeeld Tall en Thomas, 2002). Israëlische onderzoekers van het informaticaonderwijs, zoals Hazzan (2002) en Aharoni (2000) lieten zich door Skemp en zijn collega's inspireren, op zoek naar denkniveaus binnen de informatica. Ze hielden zich bezig met begrippen als berekenbaarheid en datastructuur. De mate van abstractie van een denkniveau heeft volgens hen meerdere aspecten, maar het volgens ons meest fundamentele is abstractieniveau als uitdrukking van de verhouding tussen proces en object. In deze opvatting bereikt een student een hoger abstractieniveau met betrekking tot een bepaald concept, wanneer hij of zij processen en relaties tussen objecten die verband houden met dat concept, weer als een nieuw soort objecten kan zien. We werken dit uit voor het begrip algoritme.

## Niveaus bij het begrip algoritme

Toegepast op het begrip algoritme is de volgende door ons gehanteerde vierdeling in niveaus mogelijk:

1. Het executieniveau: het algoritme is een specifieke run op een concrete specifieke machine; de benodigde tijd voor uitvoering wordt door die machine bepaald.
2. Het programmaniveau: het algoritme is een proces, beschreven door een specifieke, uitvoerbare programmeertaal; de uitvoeringstijd hangt af van de input.
3. Het objectniveau: het algoritme staat los van een specifieke programmeertaal; het wordt niet meer als proces, maar als object gezien; bij de constructie van een algoritme worden datastructuur en invariantie-eigenschappen gebruikt; meta-eigenschappen, zoals terminatie, zijn relevant en de benodigde tijd wordt beschouwd in termen van ordegrootte als functie van de input.
4. Het probleemniveau: het algoritme kan als een object worden gezien met de eigenschappen van een 'black box'; het perspectief is geworden: gegeven een probleem, welk type algoritme is geschikt? Problemen kunnen worden gecategoriseerd volgens geschikte algoritmen; een probleem heeft een intrinsieke complexiteit.

## Geen vaste niveaus

Hazzan (2002) en Aharoni (2000) onderzoeken niveaureductie. Door interviews met studenten en analyse van uitwerkingen van opgaven ontdekten ze dat studenten vaak de neiging hebben problemen op een lager abstractieniveau aan te pakken dan eigenlijk bedoeld was. Onze benadering is iets anders. We veronderstellen dat er verschillende niveaus mogelijk zijn en dat studenten in de loop van hun studie groeien in hun niveau. Bij het bereiken van een hoger niveau gaat het oude niveau niet verloren, maar wordt opgenomen in het nieuwe geheel. Een probleem kan een bepaald niveau van denken oproepen. Een student heeft wel de mogelijkheid maar niet de noodzaak om op het hoogst beschikbare niveau te reageren.

## De Eindhovense opleiding

Het algoritmeonderwijs heeft een apart karakter bij de opleiding Technische Informatica aan de Technische Universiteit Eindhoven. Lange tijd werd het curriculum gedomineerd door de traditie van de zogenaamde Eindhovense School met een fundamentele aanpak van het algoritmeonderwijs, gekenmerkt door gebruik van de wiskunde als effectief ontwerpgereedschap, door correctheid van algoritmen via constructie in plaats van verificatie achteraf en door systematisch gebruik van abstractie om complexiteit te beheersen. Recent werd de moderne ontwikkeling van software engineering, met constructie van systeemcomponenten gevolgd door het samenstellen daarvan tot grotere gehelen, in het curriculum opgenomen. Hoewel in discussie wordt de grondige methode van correctheid via constructie bij het algoritme onderwijs nog steeds gebruikt. De keuze voor het begrip algoritme als onderzoeksobject is dus een voordehandliggende, gezien het karakter van de opleiding.

## Hypothesen

- We verwachten niveaus te kunnen onderscheiden binnen de groep van bachelorstudenten.
- We verwachten dat de groep studenten in het derde jaar een hoger niveau heeft dan de groep in het tweede jaar, en die weer hoger dan de groep in het eerste jaar.
- We verwachten niveauverhoging bij een zelfde groep studenten gedurende het jaar.

- We verwachten dat het niveau van een student aan het eind van een onderwijsperiode in een algoritmevak samenhangt met de mate van succes op het tentamen over dat vak.

Tenslotte onderzoeken we of de onderwijsgevenden ook enig globaal zicht hebben op het denkniveau van hun studenten.

## Opzet

Eerst is een vragenlijst geconstrueerd en een bijbehorend scoringssysteem ontwikkeld. De vragenlijst werd binnen het academisch jaar 2003-2004 afgenomen bij de tentamens van vijf algoritmegerelateerde vakken, te weten:

- Programmarealisatie 1, op het eind van het eerste trimester van het eerste jaar (1.1.)
- Ontwerp van Algoritmen 1, op het eind van het tweede trimester van het eerste jaar (1.2)
- Ontwerp van Algoritmen 2, op het eind van het eerste trimester van het tweede jaar (2.1)
- Ontwerp van Algoritmen 3, op het eind van het derde trimester van het tweede jaar (2.3.)
- Complexiteit, op het eind van het eerste trimester van het derde jaar (3.1.).

Daarnaast werd de vragenlijst voorgelegd aan een aantal van de betrokken onderwijsgevenden, enerzijds bij collegedocenten en anderzijds bij de praktische oefeningen betrokken instructeurs en student-assistenten.

## De vragenlijst

De uiteindelijke vragenlijst bestaat uit zeven items. De lijst begint met het volgende item:

0. Geef je definitie van algoritme.

Dit wordt gevolgd door zes items in de vorm van een vraag of men het eens is met een bepaalde uitspraak (eens, oneens, eens en oneens kan beide, weet niet) plus de vraag om argumenten bij het gekozen alternatief.

1. Een algoritme is een programma, geschreven in een programmeertaal.
2. Twee verschillende programma's in dezelfde programmeertaal kunnen implementaties zijn van hetzelfde algoritme.
3. De correctheid van een algoritme is in het algemeen te bewijzen door de implementatie te testen met slim gekozen testcases.
4. Een geschikte maat om te meten hoe lang een bepaald algoritme erover doet om een bepaald probleem op te lossen, is de tijd die het kost in milliseconden.
5. De complexiteit van een probleem is onafhankelijk van de keuze van het algoritme waarmee je het oplost.
6. Bij ieder probleem is het mogelijk dat in de toekomst algoritmen worden gevonden die een ordegrootte efficiënter zijn dan de nu bekende.

Bij de vragenlijst werden gedetailleerde scoringsregels ontwikkeld om per vraag het niveau van het antwoord te meten (1, 2 of 3 bij de items 0 tot en met 4; 1, 2, 3 of 4 bij de items 5 en 6). Oorspronkelijk was de lijst langer, maar met deze verkorte lijst lukte het twee scoorders tot voldoende overeenstemming te brengen (Cohens Kappa .64). Uit de serie gescoorde antwoorden van een student werd vervolgens het gemiddelde antwoordniveau bepaald als de mediaan van de gescoorde abstractieniveaus.

## Resultaten tussen jaargangen

Een aantal studenten gaf geen of onduidelijke argumentatie bij hun antwoorden; bij

ongeveer 85% bleek een gemiddeld antwoordniveau wel te bepalen. Het bleek inderdaad mogelijk verschillende niveaus te onderscheiden, zij het beperkt. In tabel 1 worden de groepen vergeleken aan het eind van respectievelijk trimester 1.1, 2.1 en 3.1. Vrijwel alle studenten hebben een gemiddeld antwoordniveau van 2 tot 3.

Tabel 1: Abstractieniveau bij verschillende jaargroepen

| Jaargroep in trimester | Percentage studenten met niveauscore | | | | Aantal |
|---|---|---|---|---|---|
| Niveauscore | 1.5 | 2 | 2.5 | 3 | |
| Jaargroep 1 eind 1.1 | 4 | 50 | 6 | 40 | 67 |
| Jaargroep 2 eind 2.1 | | 21 | 7 | 72 | 58 |
| Jaargroep 3 eind 3.1 | | 8 | 2 | 90 | 72 |

In hogere jaren is het antwoordniveau inderdaad in het algemeen hoger (een significante rangcorrelatie Spearmanns Rho van 0.48).

## Resultaten door een jaar heen

Volgens verwachting groeide het gemiddelde antwoordniveau in de loop van een jaar bij de meeste studenten. In tabel 2 zijn de resultaten van de twee eerstejaars metingen en de twee tweedejaars metingen weergegeven. De percentages zijn gegeven van de studenten met respectievelijk een hoger, zelfde en lager gemiddeld antwoordniveau.

Tabel 2: Niveaugroei gedurende een jaar

| Jaargroep (trimesters) | % met hoger niveau | % met zelfde niveau | % met lager niveau | Aantal |
|---|---|---|---|---|
| 1 (1.1→1.2) | 48 | 44 | 8 | 36 |
| 2 (2.1→2.3) | 34 | 56 | 10 | 48 |

In beide gevallen is er sprake van significante niveauverhoging (Wilcoxon Signed Ranks Test met Z=-2.47 en -2.27 respectievelijk).

## Resultaten samenhang met tentamen cijfers

Slechts bij één tentamen was er een kleine maar significante correlatie tussen gemiddeld antwoordniveau bij de afsluiting van een vak en het tentamencijfer voor het vak en wel bij het derdejaars vak Complexiteit. Bij de andere vakken was de correlatie telkens dicht bij 0. Zie tabel 3.

Tabel 3: Correlatie tussen abstractieniveau en tentamencijfer bij de betrokken vakken

| **Vaktentamen** | **Rangcorrelatie** | **Aantal** |
|---|---|---|
| Programmarealisatie 1 | .14 | 58 |
| Ontwerp van Algoritmen 1 | .05 | 63 |
| Ontwerp van Algoritmen 2 | -.05 | 44 |
| Ontwerp van Algoritmen 3 | .09 | 72 |
| Complexiteit | .27* | 72 |
| * = significant op 0.05 (tweezijdig) | | |

## Resultaten docenten

Ruim de helft van de onderwijsgevenden betrokken bij de uitgekozen algoritmevakken, collegedocenten, instructeurs en student-assistenten, vulde ook de vragenlijst in. Dit deden ze twee keer: vanuit het perspectief van de gemiddelde student die voldoende de stof beheerste om voor het tentamen te slagen en vanuit het perspectief van de student die onvoldoende de stof beheerste. Ook de docenten scoorden op deze wijze van niveau 2 tot niveau 3 en ze scoorden als goede student een beter niveau dan als zwakke student. Er was echter geen samenhang van de hoogte van het niveau met de jaargroep en er was ook geen niveauverhoging zichtbaar bij dezelfde groep in de loop van het jaar, zoals valt af te lezen uit tabel 4. Verder gaven vooral de collegedocenten aan het een moeilijke taak te vinden. Soms namen ze voor de goede student zichzelf en konden ze voor de zwakke student helemaal geen antwoord geven.

Tabel 4: Anwoordniveaus volgens docenten

| Trimester | Onderwijsrol | Geschat niveau studenten | |
|---|---|---|---|
| | | 'Zakker' | 'Slager' |
| 1.1 | SA | 3 | 3 |
| | SA | 2 | 3 |
| | SA | 2 | 2 |
| | CD | - | 3 |
| 1.2 | CD | - | 2 |
| | I | 2 | 2.5 |
| | I | 2 | 3 |
| | I | 2 | 3 |
| 2.1 | I | 2 | 3 |
| | CD | - | 3 |
| 2.3 | I | 2 | 2.5 |
| | I | - | 3 |
| 3.1 | CD | 2 | 3 |

SA = studentassistent; CD = collegedocent; I = instructeur; - = missend of niveau niet te bepalen

## Conclusies en Discussie

Het is gelukt een vragenlijst te construeren om t.a.v. het begrip algoritme het (gemiddeld) denkniveau van studenten te meten. De betrouwbaarheid – hoe goed gemeten wordt wat gemeten wordt – hebben we op geaccepteerd niveau kunnen brengen. De validiteit – in hoeverre gemeten wordt wat bedoeld was te meten – zou nog versterkt kunnen worden. We analyseerden bij een groot aantal studenten de argumentatie bij de gekozen antwoorden van de vragenlijst. We zouden in vervolgonderzoek ook nog bij een kleiner aantal studenten zowel de vragenlijst als een langer mondeling interview kunnen afnemen om hun niveau van algoritmebegrip te peilen. Wanneer we duidelijke verbanden zichtbaar zouden kunnen zichtbaar maken tussen het niveau volgens de vragenlijst en het niveau volgens het interview, zou de validiteit daarmee nog versterkt zijn.

Vrijwel alle studenten kwamen bij hun gemiddelde beantwoording van de vragen in de range niveau 2 tot niveau 3 terecht. Mogelijk is het eerste niveau sterker aan het eind van het VWO te vinden of aan het begin van het HBO; voor een betere zichtbaarheid van het vierde niveau zou de vragenlijst moeten worden uitgebreid en zou bij masterstudenten gemeten moeten worden. Het zou ook interessant zijn de vragenlijst aan studenten van een andere universitaire informaticaopleiding voor te leggen.

Er bleek een hoger niveau te zijn in opvolgende jaargangen en er was niveauverhoging te constateren gedurende het jaar. Aangezien het gaat om correlaties moeten we voorzichtig zijn met het trekken van conclusies aangaande dat het algoritmeonderwijs ook de oorzaak zou zijn van deze niveauverhoging. Voor dergelijke conclusies zou experimenteel onderzoek nodig zijn: een opzet met vergelijkbare groepen, die zo mogelijk alleen zouden verschillen op het punt van het al of niet volgen van de betreffende vakken. Dit is praktisch lastig uitvoerbaar. De gevonden resultaten kunnen ook door andere factoren veroorzaakt zijn, bijvoorbeeld het onderwijs in het algemeen of zelfs het ouder worden op zich.

Er was onverwacht weinig samenhang tussen het antwoordniveau van een student voor onze vragenlijst en het antwoordniveau op het algoritmetentamen tezelfdertijd. Misschien hadden die docenten gelijk die vooraf al aangaven dat het bij hun vak (onderwijs in de zogenaamde Guarded Command Language voor het correct construeren van elementaire algoritmen, zie Kaldewaij, 1990) eigenlijk meer om precisie gaat dan om abstractie. Het zou ook interessant zijn cijfers voor andere vakken uit dezelfde tentamenperiode in de vergelijking te betrekken.

Bovenstaand gebrek aan samenhang tussen tentamencijfer en niveau verklaart ook deels de moeilijkheid van docenten de vragenlijst in te vullen vanuit het perspectief van de zwakke en de sterke student. Aan de andere kant was het opvallend hoe lastig docenten het vonden zich in de denkwereld van de student – vooral de zwakke student – te verplaatsen. Zonder enig zicht daarop bestaat het risico, dat - zeker bij colleges met weinig interactie - het gegeven verhaal geen aansluiting vindt. Bij de instructies is door de meer intensieve interactie de kans op aansluiting met de denkwereld van de student groter. Het zicht op die denkwereld is aan de ene kant te vergroten in de praktijk: door onderwijs met veel interactie; aan de andere kant door onderzoek, bijvoorbeeld zoals hier beschreven. Een verbreding zou zijn het identificeren van *alle* belangrijke begrippen in de informatica en het karakteriseren van de ontwikkeling die studenten daarin doormaken.

In Nederland kennen we het Utrechtse Freudenthal Instituut voor onderzoek van het wiskundeonderwijs vanuit de vakdidactiek en ook Groningen en sinds kort Heerlen moeten op dit punt niet vergeten worden; daarnaast is er onderzoek van het wiskundeonderwijs vanuit onderwijskundige en psychologische context. Meestal betreft het dan primair en secundair onderwijs, maar ook het hoger onderwijs komt langzamerhand meer binnen het gezichtsveld. Wanneer nu eindelijk het informaticaonderwijs vaste voet op HAVO en VWO gaat krijgen en er volwaardige lerarenopleidingen informatica komen, lijkt ook het moment gerechtvaardig voor het opzetten van een Instituut voor Onderzoek en Ontwikkeling van het InformaticaOnderwijs[1]. Zo'n instituut zou kunnen profiteren van onderzoek en ontwikkeling van het wiskundeonderwijs, zoals bepleit door Almstrum e.a. (2002), van onderzoek en ontwikkeling bij andere exacte vakken (Almstrum e.a., 2003), van relevant onderzoek uit de sociale wetenschappen, en last but not least van de praktische ervaringen vanuit het CODI (Consortium Omscholing Docenten Informatica) en vanuit de informaticadocenten zelf.

## Referenties


- Aharoni, D. (2000), Cogito, Ergo, Sum! Cognitive Processes of Students Dealing with Data Structures. *Proceedings SIGCSE*, Austin, blz. 26-30.
- Franquinet, R. (2004), Informatica VO uit de kinderschoenen; *Tijdschrift voor informaticaonderwijs, 13e jaargang, nummer 2*, blz. 46-49.
- Almstrum, V.L. e.a. *(*2002), Import and Export to/from Computing Science Education: The Case of Mathematics Education Research. *Proceedings ITiCSE*, Aarhus, blz. 193-194.
- Almstrum, V.L., e.a. (2003), Transfer to/from Computing Science Education: The Case of Science Education Research. *Proceedings SIGCSE*, Reno, blz. 303-304.
- Hazzan, O. Reducing Abstraction Level when Learning Computability Concepts. *Proceedings ITiCSE*, Aarhus, 2002, blz. 156-160.
- Kaldewaij, A. *Programming: The Derivation of Algorithms.* Prentice Hall International, UK, 1990.
- Tall, E. & Thomas, T. (Ed.) (2002), *Intelligence, Learning and Understanding in Mathematics; a tribute to Richard Skemp*. Post Pressed, Flaxton.


## Noot

1. Alsof het was afgesproken: op het NIOC-congres werd inderdaad de eerste stap gezet om tot een IOIO te komen.