

Mark Timmer

*Twents Carmel College, Oldenzaal, en
ELAN, Vakgroep Docentontwikkeling, UT
m.timmer@utwente.nl*

Jos Tolboom

*SLO, nationaal expertisecentrum leerplanontwikkeling
Enschede
j.tolboom@slo.nl*

Onderwijs

Computational thinking – een vooruitblik op het wiskundeonderwijs van de toekomst

Wie te maken heeft met het voortgezet wiskundeonderwijs, zal in de komende jaren ongetwijfeld op enig moment de term ‘computational thinking’ langs horen komen. Deze verzamelaar voor een verscheidenheid aan competenties op het raakvlak van wiskunde en informatica zou wel eens een behoorlijke stempel kunnen gaan drukken op het onderwijs van de toekomst. Mark Timmer (gepromoveerd theoretisch informaticus, wiskundeleraar en vakdidacticus wiskunde) en Jos Tolboom (gepromoveerd wiskundeleraar en leerplanontwikkelaar wiskunde en informatica) geven u alvast een voorproefje van wat eraan zit te komen.

Hoewel het vakgebied wiskunde zich lastig laat definiëren, heerst er algemene consensus dat wiskunde gaat over patronen, verbanden en structuren – in de praktijk veelal met het doel om bepaalde (al dan niet kwantificeerbare) vraagstukken op te lossen. Steeds vaker worden praktische problemen die met wiskundige technieken kunnen worden aangepakt dusdanig complex, dat handmatig algebraïsch-analytisch oplossen niet meer tot de mogelijkheden behoort en de computer moet worden ingeschakeld. Om dat voor elkaar te krijgen moeten we uiteraard beschikken over de technische vaardigheden om die computer daadwerkelijk aan het werk te zetten (zoals programmeren), maar voorafgaand aan het concrete programmeerwerk zal eerst goed moeten worden nagedacht over óf, en zo ja, hoe, het probleem überhaupt – eventueel ten dele – geautomatiseerd kan worden opgelost. Deze eerste stap wordt sinds enkele jaren aangeduid met de term *computational thinking* (ook wel computationeel denken genoemd) [1,6,8]. De

belangrijkste bron van het achterliggende gedachtegoed is het werk van Seymour Papert [3], dat de rol van computerondersteuning voor probleemoplossen in wiskundige situaties expliciet beschrijft.

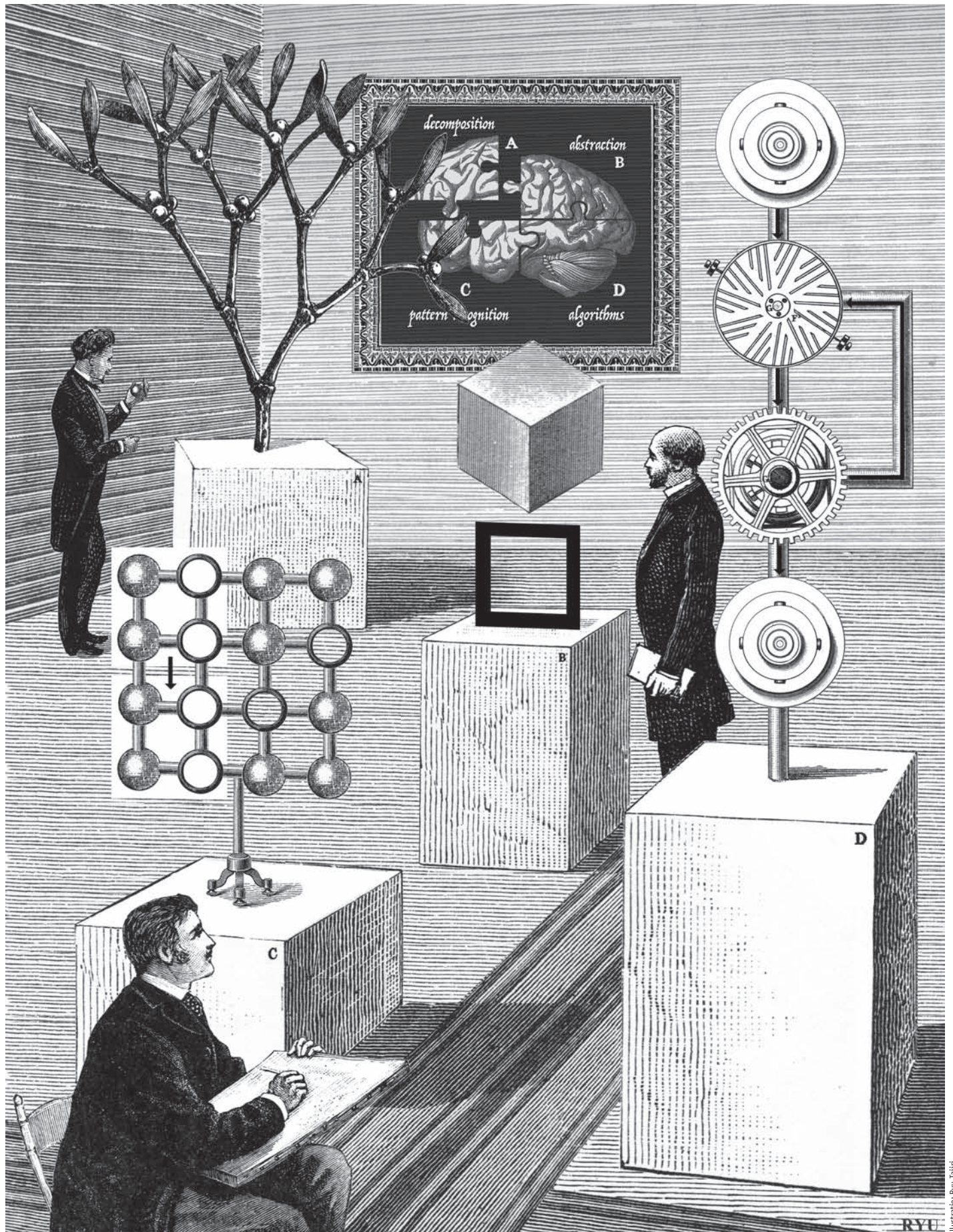
Computational thinking

Door een van grondleggers van het concept wordt computational thinking omschreven als [7]: “The thought processes involved in formulating a problem and expressing its solution(s) in such a way that a computer – human or machine – can effectively carry out.” Te denken is hierbij onder andere aan decompositie (een probleem slim in kleinere deelproblemen opsplitsen), abstractie (het weglaten van overbodige details) en algoritmiek (het formuleren van een oplossingsstrategie op een manier die door computers uit te voeren is). Ook het analyseren van en reflecteren op de efficiëntie van algoritmen (computationele complexiteit) wordt als een deelvaardigheid van computational thinking gezien [2].

Curriculum.nu, het landelijke vernieuwingsproject dat richting geeft aan de toekomst van het basis- en voortgezet onderwijs, noemt computational thinking in het ontwerpteam ‘Digitale geletterdheid’ als een van de vier pijlers van digitale geletterdheid – naast basis ICT-vaardigheid, informatievaardigheid en mediawijsheid. Hoewel de eerdergenoemde aspecten van computational thinking traditioneel vaak onder het vakgebied van de informatica worden geschaard, leent ook het vak wiskunde in het voortgezet onderwijs zich prima om leerlingen te onderwijzen in computational thinking. Het ligt voor de hand om daar gebruik van te maken, aangezien niet alle leerlingen het vak informatica volgen. Onderwijsactiviteiten gericht op computational thinking passen bovendien goed binnen het thema ‘wiskundige denkactiviteiten’, dat in het huidige wiskunde curriculum van havo en vwo al het uitgangspunt vormt voor het vormgeven van het examenprogramma in de concrete domeinen (en dat met aspecten als probleemoplossen, analytisch denken, abstraheren en modelleren ook zeker overlap vertoont met computational thinking).

Ruiken aan de grenzen van de wetenschap

Leerlingen in het voortgezet onderwijs lijken vaak weinig gevoel te hebben voor het feit dat wiskunde nog constant in ontwikkeling is. Veel van de technie-



ken die op school aan bod komen zijn al eeuwen oud, en het feit dat ze ooit door iemand zijn ‘uitgevonden’ komt niet vaak aan bod. Dit kan ervoor zorgen dat leerlingen wiskunde als een statisch gegeven gaan beschouwen, wat niet uitnodigend werkt voor een carrière in de wiskunde, in onderwijs, wetenschap of het bedrijfsleven.

Computational thinking biedt naar ons idee perspectieven om een bijdrage te leveren aan het dynamische imago van de wiskunde. Immers, als we kijken naar de grenzen van de wetenschap en het bedrijfsleven, dan ligt de uitdaging vaak minder in het daadwerkelijk programmeren van wiskundige hulpmiddelen om bepaalde problemen op te lossen, maar meer in het begrijpen en analyseren van die problemen en het herformuleren ervan op een

duddanige manier dat eraan gerekend kan worden – computational thinking dus.

Een mooie illustratie van de plaats van computational thinking aan de grens van de wetenschap vinden we in het promotieonderzoek van de eerste auteur. Onder de noemer ‘probabilistic model checking’ deed hij onderzoek naar efficiëntieverbetering van technieken om automatisch bepaalde kwantitatieve eigenschappen van protocollen of systemen te berekenen [4]. Dergelijke onderwerpen uit de theoretische informatica vragen om een mix van wiskundige modelleerconcepten om de werkelijkheid om te zetten in een handzaam formaat (zoals procesalgebra), wiskundige bewijstechnieken om te kunnen redeneren over de correctheid van efficiëntieverhogende herschrijfgeregels, en technische vaardigheden om dit alles in de praktijk te brengen in

de vorm van een implementatie waarmee miljoenen toestanden van een systeem in korte tijd doorgerekend kunnen worden. Als we leerlingen alvast een kijkje kunnen geven in de keuken van dergelijke projecten, door hen kennis te laten maken met wiskundige denkwijzen en technieken die benodigd zijn om op zo’n manier wiskunde in de praktijk te kunnen brengen, dan hopen we ze te kunnen enthousiasmeren om zelf ook met wiskunde aan de slag te willen gaan aan de grenzen van de wetenschap. Er is immers nog genoeg te doen.

Ontwikkeling van onderwijsmateriaal

Aan concrete plannen voor een inbedding van computational thinking in het wiskunde-curriculum van de toekomst wordt momenteel nog gewerkt, maar het ligt voor de hand dat alle wiskundedocenten (in ieder

Een voorbeeld van lesmateriaal over computational thinking

In de opdracht van Van der Meulen gaan leerlingen onder andere aan de slag met het plannen van taken die sequentieel op één machine moeten worden uitgevoerd. Hierbij dienen ze rekening te houden met de lengte van iedere taak (*processing time p*), het tijdstip waarop de taak beschikbaar komt (*release date r*) en het uiterste moment waarop de taak klaar dient te zijn (*due date d*). Zie Tabel 1 voor een illustratie van dergelijke gegevens voor een drietal taken.

taak:	1	2	3
<i>p</i>	5	6	5
<i>r</i>	0	4	2
<i>d</i>	15	11	12

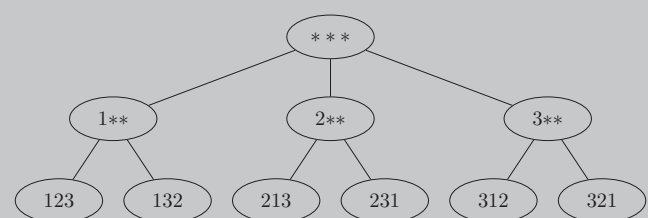
Tabel 1 Drie taken.

Op basis van dit soort informatie moet bedacht worden in welke volgorde de taken het best kunnen worden uitgevoerd, waarbij eerst ook nagedacht wordt over een optimalisatiecriterium. Het blijkt immers niet mogelijk om alle taken voor hun deadline af te krijgen. Te denken is aan het minimaliseren van het aantal taken dat te laat is, het minimaliseren van de totale vertraging van alle taken die te laat zijn of het minimaliseren van de maximale vertraging van de taken die te laat zijn. Via realistische bedrijfsscenario’s wordt aannemelijk gemaakt dat hier geen eenduidige beste keus in is.

Uiteindelijk gaan leerlingen aan de slag met de maximale vertraging, aangeduid met L_{\max} (waarbij de L komt van ‘lateness’). In het bovenstaande voorbeeld geldt bijvoorbeeld voor volgorde 123 dat $L_{\max} = 4$, en voor de optimale volgorde 321 geldt $L_{\max} = 3$. Leerlingen worden aan het denken gezet over de hoeveelheid werk om alle mogelijke volgordes door te rekenen, wat bij drie taken vanzelfsprekend prima te doen is, maar bij twintig taken al astronomisch veel tijd zou kosten. Leerlingen

gaan daarom aan de slag met heuristieken, waarbij op een efficiënte wijze een oplossing wordt gevonden die hopelijk (bijna) optimaal is, maar waarvan dat niet gegarandeerd kan worden. Het nadenken over situaties waarin de heuristiek nu juist niet tot de beste oplossing leidt, is hierbij een mooie wiskundige denkactiviteit.

Nadat leerlingen hebben ontdekt dat complete enumeratie al snel ondoenlijk wordt en heuristieken niet altijd tot de perfecte oplossing leiden, komt ‘branch and bound’ om de hoek kijken. Voor een uitgebreide uitleg over deze wiskundige techniek en de inbedding daarvan in de opdracht over scheduling verwijzen we naar [5]. Kort gezegd komt het erop neer dat alle mogelijke oplossingen – dat wil zeggen, alle mogelijke volgordes van de drie taken – in een boomstructuur geplaatst worden, waarbij iedere vertakking meer vastlegt betreffende de volgorde van de taken (zie Figuur 1). Door gebruik te maken van handige ondergrenzen kunnen we de boomstructuur doelgericht doorlopen en (hopelijk) vele takken wegstrepen zonder dat alle oplossingen die erin bevat zijn hoeven te worden doorgerekend. Op die manier kan het aantal mogelijke volgordes waarvan de L_{\max} berekend moet worden vaak drastisch worden beperkt, zonder dat optimale oplossingen over het hoofd worden gezien. Een mooi voorbeeld van een slimme aanpak, waarbij wiskundig denken leidt tot een algoritmische wijze om tot oplossingen te komen, die eenvoudig door de computer kan worden uitgevoerd en waarbij we met zekerheid tot de gewenste oplossing komen.



Figuur 1

geval in het havo/vwo) over een jaar of tien geacht worden hun leerlingen computationeel te prikkelen. Hoewel hiermee de vorm van het wiskundeonderwijs wellicht iets verandert, blijft het onderliggende doel gelijk: leerlingen leren denken. Op dit moment wordt er al op kleine schaal geëxperimenteerd met dergelijk lesmateriaal waarin computational thinking centraal staat.

In zijn afstudeerscriptie voor de opleiding tot eerstegraads docent wiskunde heeft Joris van der Meulen bijvoorbeeld een werkmiddag ontwikkeld voor 5 vwo wiskunde B, waarin het algoritme-ontwerp-paradigma 'branch and bound' uit de doeken wordt gedaan [5]. Op basis van opdrachten omtrent de optimalisatie van scheduling (taakplanning) gingen leerlingen aan de slag met algoritmen, heuristieken en computationele complexiteit – zie het kader op de vorige pagina voor een uitgebreidere toelichting. Zowel de docenten als de leerlingen waren enthousiast over eens een hele andere manier van werken bij wiskunde; niet alleen maar oefenen met het oplossen van concrete problemen, maar ook op een hoger abstractieniveau nadenken over hoe we dat eigenlijk aanpakken en waarom dat efficiënt en/of optimaal is. Activiteiten zoals het laten bear-

gumenteren waarom een reeds gevonden oplossing van een schedulingprobleem onder bepaalde voorwaarden gegarandeerd optimaal is, vragen behoorlijk wat van leerlingen en zijn uitermate geschikt voor het aanleren van een wiskundige houding. Leerlingen bleken dit soort taken, overeenkomstig met onze verwachtingen, lastig te vinden, maar velen zagen er ook het nut van in en werden er enthousiast van.

Het Nationaal Regieorgaan Onderwijsonderzoek (NRO) subsidieert de komende drie jaar praktijkgericht onderzoek naar computational thinking onder de titel 'Computationeel denken en wiskundig denken: digitale geletterdheid in wiskundecurricula' [9]. Vanuit een consortium van de Universiteit Utrecht, de Radboud Universiteit, SLO en een vijftal vo-scholen wordt er onderzocht welke aspecten van computational thinking goed passen bij wiskundig denken en in de bovenbouw van het vwo aan bod zouden kunnen komen. Ook wiskundeleraaropleidingen, zoals die van de Universiteit Twente, dragen bij. Het einddoel is een verzameling van empirisch gevalideerde leeractiviteiten voor zowel wiskunde A als wiskunde B, een aantal instrumenten voor het toetsen van de uitkomst van dergelijke leeractiviteiten en een handleiding voor

docenten om hen te ondersteunen bij het onderwijzen van computational thinking, gekoppeld aan wiskundige denkactiviteiten en alles op basis van functionele inzet van ICT in de leermiddelen.

Concluderend

Computational thinking overkoepelt een grote hoeveelheid aan vaardigheden en denkwijzen die nu al erg belangrijk zijn in de wetenschap en het bedrijfsleven, en die in de toekomst vermoedelijk alleen maar belangrijker worden. Het is wat ons betreft dan ook niet vreemd dat er vanuit verschillende hoeken geprobeerd wordt om hier meer van in het (voortgezet) onderwijs te krijgen. De wiskundeles lijkt, naast de informaticalessen, een uitgelezen plek om leerlingen te prikkelen en verder te krijgen.

Mocht u vanuit uw werkveld interessante ideeën hebben omtrent het inpassen van computational thinking in het wiskundecurriculum, schroom dan niet om contact op te nemen met een van de auteurs. ☺

Dankwoord

De auteurs danken Paul Drijvers, Robbert Fokink en Joris van der Meulen voor het geven van feedback op een eerdere versie van dit artikel.

Referenties

- 1 S. Grover en R. Pea, Computational Thinking in K-12; A Review of the State of the Field, *Educational Researcher* 42(1) (2013), 38–43.
- 2 ISTE/CSTA, Operational definition of computational thinking for K-12 education (2018), <https://c.ymcdn.com/sites/www.csteachers.org/resource/resmgr/CompThinkingFlyer.pdf>.
- 3 S. Papert, *Mindstorms: Children, Computers, and Powerful Ideas*, Basic Books, 1980.
- 4 M. Timmer, *Efficient Modelling, Generation and Analysis of Markov Automata*, proefschrift, Centre for Telematics and Information Technology (CTIT), Universiteit Twente, 2013.
- 5 M. Timmer en J. Van der Meulen, Computational thinking in vwo 5: Een werkmiddag over 'branch and bound', *Euclides* 94(3) (2018), 14–17.
- 6 J.M. Wing, Computational thinking, *Communications of the ACM* 49(3) (2006), 33–35.
- 7 J.M. Wing, Computational thinking benefits society, *40th Anniversary Blog of Social Issues in Computing*, 2014. Geraadpleegd van <http://socialissues.cs.toronto.edu/2014/01/computational-thinking>.
- 8 J.M. Wing en D. Stanzione, Progress in computational thinking, and expanding the HPC community, *Communications of the ACM* 59(7) (2016), 10–11.
- 9 <https://www.nro.nl/nro-projecten-vinden/?projectid=40.5.18540.130>.