

De kunst van het programmeren

Dirk van Delft

NRC Handelsblad*
 nrc@nrc.nl
<http://www.nrc.nl/>

28 maart 1996

Abstract

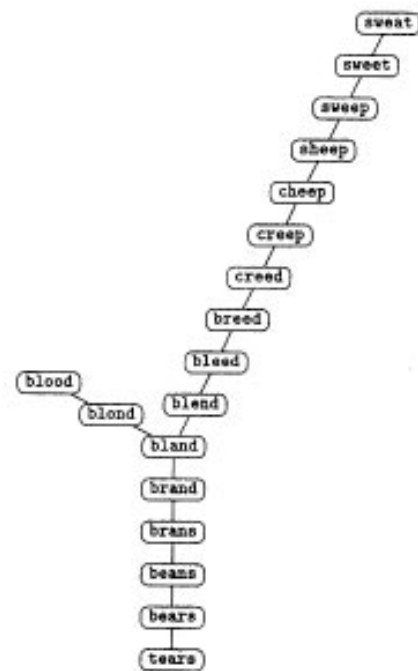
Sinds 1962 werkt Donald Knuth aan een informaticabijbel die over twintig jaar af moet zijn. Tussendoor ontwierp hij een tekstverwerker met eigen typografie en legde hij een Mona Lisa uit dominosteentjes.

In 1952 meldde Donald Knuth, toen veertien jaar oud, zich twee weken schoolziek, zogenaamd met maagpijn. In werkelijkheid ploegde de aanstaande informatica-pionier zich stiekem op zijn kamertje in Milwaukee door een woordenboek, op zoek naar woorden die zich uit de letters van 'Ziegler's Giant Bar' lieten samenstellen. De lokale fabrikant had deze wedstrijd uitgeschreven. Donald kwam op een totaal van 4.500, tweeduizend meer dan de jury, en dat terwijl hij vergeten was de apostrof te benutten. Toen zijn ouders de gedrevenheid van hun zoon opmerkten, hielpen ze de lijst uittikken en het resultaat was dat de hele klas een reuzenreep kreeg, en de school een tv-toestel.

"Ik hou van taal", zegt Knuth, op bezoek in Amsterdam ter ere van het 50-jarig bestaan van het CWI, het Centrum voor Wiskunde en Informatica. "De eerste keer dat ik de krant haalde was als vierjarige boekenwurm, opgesloten in de bibliotheek omdat ik de tijd was vergeten. Als kind schreef ik radioshow's en op mijn twaalfde zette ik thuis een schoolkrant in elkaar, met zelfbedachte puzzels. Woorden hebben zowel betekenis als vorm, en juist de combinatie geeft me plezier. Zo heb ik twintig jaar lang vijf-letterwoorden verzameld — bij 5.757 ben ik gestopt, eigennamen als Knuth liet ik achterwege — en noteerde ik wanneer en waar romanpersonages elkaar in *David Copperfield* of *Anna Karenina* troffen. In mijn informaticaboeken, waar ik duffe voorbeelden per se wil vermijden, maak ik van die lijsten dankbaar gebruik."

Zoals in de *Stanford GraphBase*, een verzameling 'leesbare' computerprogramma's en datasets die Knuth als hoogleraar informatica aan Stanford University (Californië) in 1994 publiceerde — uitgeroepen tot het beste informaticaboek van dat jaar. Een 'graaf' is in de wiskunde een verzameling 'knooppunten' die op zekere manier met elkaar in verbinding staan: de afstandstabel van de Europese hoofdsteden, of een sociogram van de ministerraad. Knuth: "Grafen zijn speelgoed voor de informaticus. Die zoekt in mijn corpus aan vijfletterwoorden de kortste weg van *words* naar *graph* als steeds één letter anders mag. Het

gaat dan om het optimum bij een zeer groot aantal mogelijkheden. Om dat snel te vinden heb je efficiënte algoritmes nodig, zoals dat van Dijkstra uit 1959. Dat leidt op een hedendaagse computer in een vloek en een zucht tot de serie *words-wolds-golds-goads-grads-grade-grape-graph*. Overigens lukt het niet altijd, waar *bares* en *cores* in de 'buren' zwemmen — beiden 25 — hebben *ocean*, *below* of *music* er geen een."

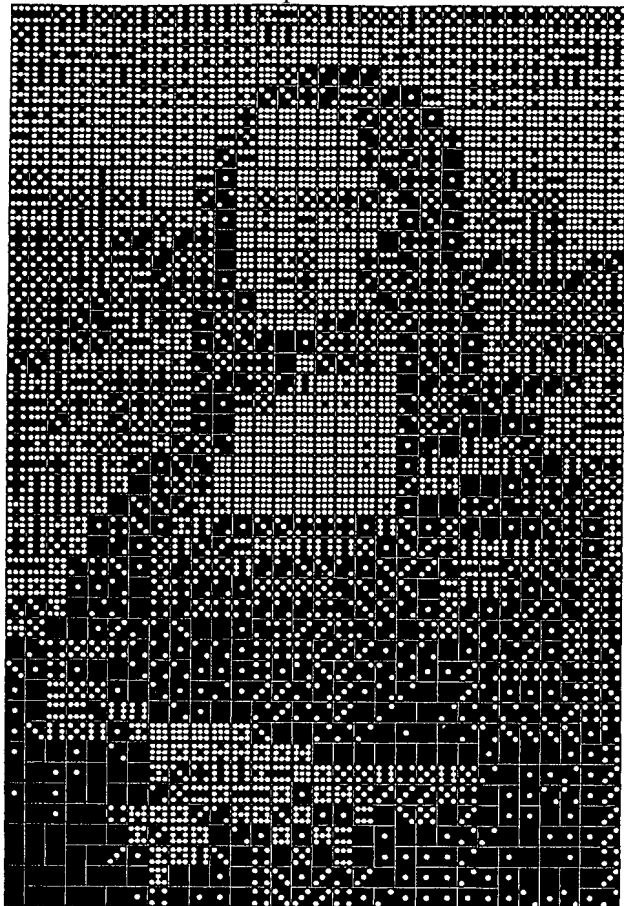


Figuur 1: De dataset 'words.dat' uit de *Stanford GraphBase* bevat 5.757 vijf-letter-woorden. Het programma 'ladder' zoekt 'buren' die in één letter verschillen. Het voorbeeld toont hoe de woorden *blood*, *sweat* & *tears* (naar een popgroep uit de jaren zestig) in zo min mogelijk stappen in elkaar overgaan.

Knuth heeft met zijn *Stanford GraphBase* een standaard willen zetten. "Een algoritme kan altijd beter, één bril-

*Bijlage 'Wetenschap & Onderwijs' blz. 3. Het artikel is in elektronische vorm via docu@nrc.nl bij het NRC Handelsblad verkregen. Copyright ©NRC Handelsblad.

jant idee en het programma loopt duizend keer sneller. Maar rekentijd zegt weinig als computers verbeteren en uitgangsgegevens per gebruiker verschillen. Mijn Stanford GraphBase ondervangt dit door standaard-datasets en -programma's te bieden die iedereen mag gebruiken, zolang hij er maar niet in knoeit. Eindelijk beschikt Tibet over dezelfde digitale Mona Lisa als Californië en kunnen algoritmes tenminste eerlijk worden vergeleken. Daarmee is de weg is vrij voor experimentele computerwetenschap, voor reproduceerbare proeven met uitkomsten die hun geldigheid behouden als de computer verandert."



Figuur 2: De Mona Lisa, opgebouwd uit 12 dominospelen (met elk 55 steentjes, van de dubbele nul tot de dubbele negen). De database 'lisa.dat' uit Knuth's Stanford GraphBase bevat een gedigitaliseerde versie van Da Vinci's beroemdste schilderij, met 360x250 pixels wier grijstint is vastgelegd in een cijfer tussen 0 (zwart) en 255 (wit). Ook in de Stanford GraphBase een programma dat uitgaande van zes of twaalf dominospelen, waarvan alle steentjes moeten worden gebruikt, een Mona Lisa opbouwt. Het minimaliseert de som van de kwadraten van de verschillen tussen de pixelwaarde uit 'lisa.dat' met de bijbehorende 0 of 255 van de dominosteentjes: 'assignment' in informatiecargon.

Vergaderziekte

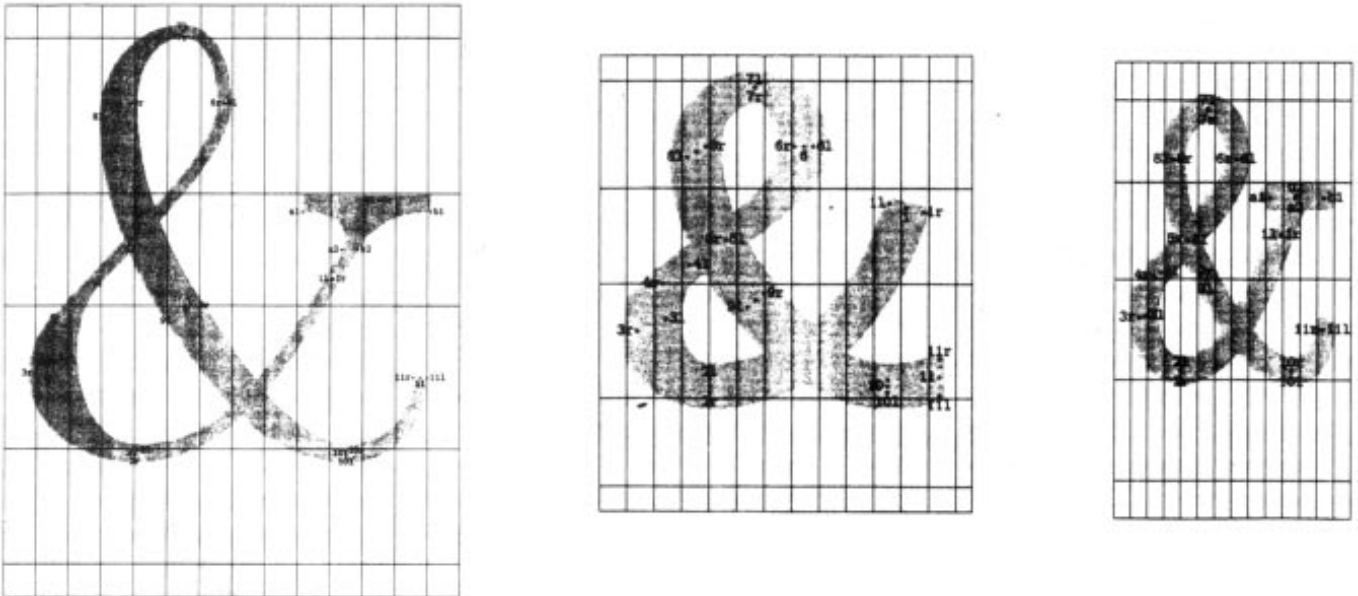
Vorig jaar ging Knuth op 56-jarige leeftijd met vervroegd emeritaat om zich, bevrijd van papierwerk en academische vergaderziekte, volledig aan de voltooiing van zijn levenswerk te kunnen wijden: *The art of computer programming*.

De oorsprong van dat project, een informaticabijbel waarvan tot nu toe drie delen zijn verschenen (met vertalingen in het Roemeens, Hongaars, Spaans, Chinees, Russisch en Japans), ligt in 1962. In dat jaar vroeg uitgever Addison-Wesley Donald, toen tweedejaars op Caltech (Pasadena), of hij zijn kennis van *compilers* (die hogere computertalen als *basic* of *fortran* vertalen naar 'machinetaal') niet te boek wilde stellen. "Toen ik die dag thuiskwam maakte ik een kladje met een indeling in twaalf hoofdstukken", herinnert Knuth zich. "Ik was net getrouwd en had geen idee van hetgeen ik me op de hals had gehaald. Programmeren lag me, tijdens een vakantiebaantje in mijn eerste jaar mocht ik 's nachts op een computer en was ik, moederziel alleen, direct verslaafd. Eerst had ik natuurkunde zullen doen, maar ik had een gloeiende hekel aan solderen. Informatica was nieuw."

The art of computer programming dijde spoedig uit. Behalve compilers wilde Knuth ook 'rangschikken' (*sorting*) behandelen, en algoritmes en optimalisering. In 1965 leverde hij een eerste manuscript in van 3.000 vel. "Ik had me er zwaar op verkeken. In overleg met de uitgever besloot ik tot zeven delen, met de compilers op het eind. Helaas is er na deel 3, verschenen in 1973, van alles tussengekomen. Intussen is het vakgebied sterk gegroeid en is revisie onvermijdelijk. Deel 4, over algoritmes, heb ik moeten splitsen in 4A, 4B en 4C. Als ik dat van tevoren geweten had, was ik er nooit aan begonnen. Nu kan ik niet meer terug, ook al omdat mijn ideeën door niemand anders fatsoenlijk zijn opgeschreven."

Veel meer dan door de Stanford GraphBase is *The art of computer programming* vertraagd door Knuth's activiteiten op het gebied van tekstverwerking en typografie. "Halverwege de jaren zeventig, toen het lood in de drukkerswereld plaats maakte voor offset, merkte ik dat de kwaliteit van de herdrukken van mijn boeken achteruit holde. Typografen hadden nauwelijks aandacht voor de speciale symbolen die in de exacte wetenschappen worden gebruikt en als drukkerszoon — er vloeit inkt door mijn aderen — leed ik daaronder. Ik besteedde de grootste zorg aan mijn boeken en wilde dat ze er mooi uitzagen. Toen ik in de gaten kreeg dat offsetmachines met digitale technieken werkten, wist ik dat er een taak voor mij was weggelegd."

Die taak resulteerde in het tekstverwerkingsprogramma \TeX (tau epsilon chi, naar de eerste letters van het Griekse woord *techne* dat techniek en kunstvaardigheid betekent) en het bijbehorende typografiesysteem METAFONT. Het geheel draait op praktisch iedere computer en is gedocumenteerd in vijf handleidingen en programmabeschrijvingen, bij elkaar zo'n 2.600 bladzijden. Bijna iedere wis- en natuurkundige schrijft zijn artikelen in \TeX . Knuth: "Ik begon in 1976 en dacht er met een jaar te zijn, het werden er tien. Ik houd niet van half werk. Ik geef om mooie curves, als scholier mocht ik graag met grafieken spelen. Die liefde keert in mijn typografie terug. Esthetische vormen in wiskunde vangen, daar gaat het me om. Ik herken mijn letters. Gisteren nog zag ik in de Leidsestraat een tram met Metafont."



Figuur 3: Voorbeeld van typografie volgens het computerprogramma Metafont. Ieder symbool is gekarakteriseerd door 62 parameters die aan of uit staan (zo dwingt monospace alle tekens dezelfde breedte op) dan wel bepaalde waardes aannemen (dikte van de stok, diepte van de komma).

Uitgangspunt voor Metafont was de ‘excellente’ typografie van (oudere) Addison-Wesley boeken, de *Proceedings* van de Koninklijke Nederlandse Akademie van Wetenschappen en het Zweedse tijdschrift *acta mathematica*. Ieder symbool, van gedachtenstreepje tot integratieteken, specificerde Knuth middels 62 variabelen. Iedere lettergrootte, ieder font (lettertype) kreeg zijn eigen instellingen. Eenzelfde onderliggende wiskundige structuur garandeert een visuele eenheid die bij handmatig ontwerpen onhaalbaar is. De gebruiker speelt met de knoppen, ziet het resultaat op scherm en beseft hoe moeilijk het is als amateurtypograaf een acceptabel alfabet te ontwerpen.

Ook T_EX zit vol geavanceerde wiskunde. Zo maakt het gebruik van een intelligent afbreekprogramma, dat niet aan één taal gebonden is, en het vult ingevoerde tekst niet per regel uit, maar per volledige alinea. Om dat te bereiken heeft Knuth een systeem bedacht dat afbrekingen of ‘overdreden’ spatiering vertaalt in ‘kosten’. Per alinea rekent T_EX van *alle* uitvoervarianten uit hoe duur ze zijn. De goedkoopste oplossing — een optimaliseringsprobleem waarvoor een Hongaars algoritme is geïmplementeerd — wint. Knuth: “Het kan zijn dat T_EX toch voor die lelijke afbreking kiest omdat *juist daardoor* nog lelijker wit op andere plaatsen wordt vermeden. Ik heb het systeem op alinea’s uit *Time* uitgeprobeerd en het verschil is tamelijk dramatisch.”

Een pagina per dag

Bevrijd van promovendi, congresbezoek en verplichte colleges richt Knuth al zijn energie op het vervolg van *The art of computer programming*. Dagelijks zit hij twee uur in de bibliotheek, tientallen artikelen tot zich nemend. Een e-mailadres heeft hij niet meer, wel een *homepage* op het world wide web, “een uitvinding invloedrijker dan de pil”. Ontspanning vindt hij achter het pijporgel (812 pijpen, 16 registers, 3 klavieren, gestemd volgens de regels van de Noord-Duitse barok) dat hij als zoon van een Lutheraans organist in zijn woning in Stanford heeft laten bouwen. Hij componeert en schreef in 1974 de novelle *Surreal numbers*, over een nieuwe manier om getallen te construeren, de enige keer dat een belangrijke wiskundige ontdekking eerst als fictie naar buiten kwam en pas daarna als artikel.

Daarvoor ontbreekt nu de tijd, *The art of computer programming* moet af. Knuth: “Een schrijftempo van een pagina per dag, dat is het streven. Concentratie is het sleutelwoord. Ik zal me moeten beheersen, alleen de tijdloze ideeën op schrift vastleggen. Informatica is zo opwindend dat ik het nauwelijks kan opbrengen naar bed te gaan. Twintig jaar heb ik nog te gaan, als het klaar is ben ik 77, laten we hopen dat ik het red. Kon ik de ontwikkelingen zolang maar stilzetten.”

$$\int_{-\infty}^{\infty} \left(\frac{\text{RAVEN}}{\text{Πάλλας}} + \heartsuit_{\tau\epsilon\lambda-\tau\alpha\lambda} \right) d\varphi \longrightarrow \left\{ \begin{array}{c} \text{THE} \\ \text{END} \end{array} \right\}$$