

Over de toekomst van Computer Science.

Het centrale probleem van Computer Science vandaag is, op welke wijze we wat voor machines zouden willen gebruiken. Vandaag is dit probleem nog open: doordat we nog niet weten, hoe we zouden willen, dat een "gebruiksaanwijzing" er uitzag, komen er machines op de markt, waarvan het een probleem is, hoe je ze gebruiken moet. Maar het is niet alleen dat ik hoop, dat dit laatste probleem een bijzonder vluchtige taakstelling zal blijken, ik verwacht zulks ook. Binnen tien jaar verwacht ik voldoende (zelf)kennis van "de Programmeur", dat wij een duidelijk beeld hebben wat voor machines goed bruikbaar zijn; tegelijkertijd zullen we dan weten, hoe we deze machines willen gebruiken. Zodra het niet meer de machines zijn, die de problemen stellen, is naar mijn smaak Computer Science klaar.

De tijdsperiode van tien jaar voor de voltooiing van Computer Science treft wellicht als bijzonder kort (en nadat het plaatsgevonden heeft, duurt het waarschijnlijk nog wel een kwart eeuw, voordat de voltooiing tot iedereen is doorgedrongen!); ik heb er echter de volgende gronden voor.

- 1) sinds dit onderzoek grotendeels gedreven wordt door de analyse van op wat voor wijze een mens betrouwbaar werken kan, is duidelijk een in de tijd vrij constant element geïntroduceerd: ongeacht alle gepraat over de kloof tussen de generaties is de mens, als je een stap terugdoet, een vrij constant gegeven.
- 2) er wordt door allerlei mensen bewust aan gewerkt en er wordt heel duidelijke voortgang geboekt.
- 3) nu is het nog zo, dat een nieuwe hardware technologie oude kennis en vaardigheden op slag obsoleet kan maken en een hele klasse nieuwe problemen stellen kan: de hardware jongens maken een gadget, waarvan je opnieuw niet weet, hoe je het in hemelsnaam op een ordelijke manier gebruiken moet. Over tien jaar verwacht ik een zo duidelijk inzicht in onze mogelijkheden en beperkingen, dat hierdoor metterdaad, ook wanneer zich een nieuwe technologie aandient, het hardware ontwerp gestuurd kan worden tot een bruikbaar apparaat. Software ontwikkeling wordt nl. ook voor de fabrikant van machines een zo levensgroot probleem, dat ik me niet kan voorstellen dat men zich nog veel langer veroorloven kan hardware eerst autonoom te ontwikkelen en vervolgens als taakstelling aan de software over te dragen.

Hoe groot is het vak, wanneer het over tien jaar klaar is? Ik heb geen maat om dat uit te drukken, maar ik zou willen zeggen: klein, althans inkrimpend. Het bewerkstelligen van deze inkrimping is nl. de taakstelling van zijn ontwikkeling! (Uit de VS hoor je -overigens typisch Amerikaanse!- geluiden als: er zullen steeds meer probleemgebonden talen komen, er zullen dus heel veel compilers gemaakt moeten worden en het is dus nodig, dat we nog veel beter leren en doceren, hoe je compilers maakt. Compilerontwikkeling is in die gedachtegang een hoeksteen. Maar als je een pas terugdoet, constateer je dat wanneer compilerbouw een probleem is, dit grotendeels te wijten is aan de taal of de machine (of aan beide, c.q. de combinatie van de twee) en ik verwacht over tien jaar het inzicht waaraan bv. taal en machine getoetst kunnen worden, zodat compiler constructie geen groot probleem meer hoeft te zijn. Het is de functie van de ontwikkeling om onze huidige moeilijkheden waar mogelijk als "self-inflicted pains" te ontmaskeren, in plaats van nieuwe pijnstillende middelen te zoeken!)

{Terzijde: ik merk, dat het erg natuurlijk komt om "ontwikkeling" van een vak onmiddellijk met kwantitatieve "groei" te associëren, zo natuurlijk, dat ik met de vorige paragraaf bijzonder ben ingenomen.}

De tegenwerping is natuurlijk onmiddellijk: "Toegegeven, dat het de functie van inzicht is, om specifieke kennis overbodig te maken. Maar waarom durf je dan te hopen, dat het juist Computer Science zal zijn, waar deze inkrimping geëffectueerd zal worden, waar het nodige inzicht groeien zal?".

Ik kan hierop een serie heel persoonlijke antwoorden geven, zoals "Ik moet in naam van mijn arbeidsvreugde wel hopen, want als ik het niet hoopte, zou het vuur waarmee ik me voor Computer Science inzet, rap gedoofd zijn en dat vuur is me te dierbaar.", maar ik wil proberen een wat onpersoonlijker antwoord te geven.

Op grond van hun versatiliteit en macht worden computers voornamelijk geapprecieerd als gereedschap en vanwege hun nog hoge prijs, waar we begrijpelijkerwijs tegenaan hikken, is programmeren in veler ogen primair een optimaliseringsopgave, nl. een minimalisering van het gebruik. Naarmate machines goedkoper en machtiger gaan worden, zal de schoen minder vaak minder hard knellen en zullen efficiencyoverwegingen minder vooropstaan. Naarmate machines evenwel machtiger worden zal de mensheid evenredig ambitieuzer worden in hun toepassingen en deze schaalvergroting zal de taak van de programmeur, die zijn werkterrein immers vindt in het spanningsveld tussen (exploderende) wensen en mogelijkheden, meer en meer maken tot een intellectuele uitdaging van de eerste orde. Dat is het al, maar de eerste ronde hebben we, doordat we de uitdaging onvoldoende onderkend hebben, verloren! Het is een hoopvol teken, dat deze onderschatting erkend begint te worden (zij het, toegegeven, schoorvoetend en beslist niet door iedereen).

In een aantal opzichten lijkt programmering, wanneer de uitdaging eenmaal onderkend is, het ideale gebied om de uitdaging te aanvaarden en met (zo je rot wil zijn: goedkoop) succes te beantwoorden:

- 1) het is, als wiskunde, een abstracte wetenschap, die zo schoon kan zijn als je hem opzet, in tegenstelling tot alle natuurwetenschappen, die een overweldigende hoeveelheid duistere observaties moeten ordenen;
- 2) in relatie tot andere takken van de wiskunde is de conceptuele basis van de programmeur doodsimpel: alle moeilijkheden van het continuum, oneindigheid, keuze-axioma's e.d. worden de programmeur bespaard en je kunt programmeren dan ook beschouwen als een "beschermd oefenterrein" par excellence;
- 3) van het feit, dat elk programma ook nog werken moet, gaat een constante, corrigerende werking uit;
- 4) het veld ontleent zijn rijkdom niet aan de smalheid van zijn basis, maar aan de macht van de feitelijk aanwezige apparatuur, die nauwelijks een bovengrens oplegt aan de "sophistication" van wat je met de machine doet;
- 5) wat je met de huidige machines al doen kunt, is al zo enorm "groot", dat de behoefte aan een heel bewuste methodologie zich onmiskenbaar presenteert;
- 6) en dit is een praktische kwestie, er is zoveel mee gemoeid, de maatschappelijke pressie is zo sterk, dat dit gebied niet links kan blijven liggen.

Het is daarom, dat ik vol goede moed ben. Het is mede daarom, dat ik verwacht, dat wij de betekenis van rekenautomaten schromelijk zullen onderschatten zolang wij ze blijven appreciëren in hun hoedanigheid van gereedschap, hoewel ze als zodanig het aangezicht van onze maatschappij veranderen; in hun hoedanigheid van intellectuele uitdaging verwacht ik van hen een veel dieper gaande invloed op onze cultuur.