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.

AANSCHOUWELIJKE COMPLEXITEITSTHEORIE

presentatie voor NIOC 2015

Jan Terlouw (Groningen)

beknopte beschrijving

Een computerprogramma moet in de praktijk aan verschillende eisen voldoen, sowieso aan de volgende twee:

1. het moet een taak verrichten zoals we dat (voor de relevante inputs) willen en

2. het moet dat ook snel genoeg doen.

De *complexiteitstheorie* houdt zich bezig met het tweede criterium en kent een grote veelzijdigheid. Probleemstelling van de presentatie:

"Hoe zijn de basisbeginselen van de complexiteitstheorie geschikt uit te leggen aan een publiek van HBO-studenten informatica?"

Commentaar van de kant van het NIOC-publiek op mijn desbetreffende ideeën zal welkom zijn. Hierbij denk ik met name, maar zeker niet alleen, aan commentaar van HBO-docenten onder dat publiek.

Wanneer er meer ruimte beschikbaar zou zijn voor de presentatie, dan zouden er systematisch, en alternerend, twee sporen uitgewerkt worden: **A**. een spoor van een nadere recapitulatie voor het NIOC-publiek van de hierboven bedoelde basisbeginselen; **B**. een spoor van een nadere bespreking van hoe die beginselen (mogelijk) nader uit te leggen zijn aan het beoogde HBO-publiek.

De ruimte is echter beperkt en reeds wat spoor A betreft zal er sprake moeten zijn van een "zeven mijls laarzen"-benadering, leunend op behoorlijke (maar niet onredelijke) aannames over een reeds bij het NIOC-publiek aanwezige voorkennis, hetgeen intussen zeker niet uitsluit dat NIOC-bezoek(st)ers met minder voorkennis ook nog het hunne kunnen meenemen van de presentatie. In relatie tot spoor B moet sowieso verondersteld worden dat een uiteindelijke uitleg voor het HBO-publiek plaatsvindt in het kader van een langere cursus, bestaande uit meerdere sessies, met genoeg ruimte voor extra inkledingen met: motiveringen, toelichtingen, illustraties, voorbeelden, etc., en uiteraard ook met opgaven.

De keuze vooraf bij deze voordacht is welbewust geweest om de nadruk te leggen op 1. de thematiek van **NP-compleetheid** (waarover verderop hieronder meer) en op 2. een desbetreffend didactisch hulpmiddel, namelijk een bepaald **bordspel** (waarover meer in detail tijdens de presentatie zelf).

De hierna volgende uiteenzetting bestaat uit **drie delen**. Expositorisch is hier in feite sprake van een vermenging van de genoemde sporen A en B. Er is echter steeds in gedachten te houden dat er voor het beoogde HBO-publiek nadere inkledingen nodig zullen zijn --- en, zo is de claim, *ook mogelijk zullen zijn.* Nadere invullingen tijdens de NIOC-presentatie zelf volgen voor zover die daarvoor de ruimte laat. In ieder geval gebeurt er tijdens die presentatie wel wat meer, ook creatief, dan in deze voorliggende beschrijving.

Deel I: voorbereidende basis.

De startmotivering die aan het begin van deze tekst gegeven is (in termen van eisen voor een computerprogramma), mag als duidelijk beschouwd worden. Tegelijk rijzen er verschillende vragen, zoals de volgende: *"Wat wordt precies (of preciezer) onder "de snelheid" van een computerprogramma verstaan?"* en: *"Wanneer is een computerprogramma "snel genoeg" of, meer in technische termen,* "efficiënt genoeg" *te noemen?"*

De snelheid van een computerprogramma wordt gemeten in termen van het aantal rekenstappen. Dat aantal zal in het algemeen afhankelijk zijn van de input x of van de --- volgens redelijke maatstaven gemeten --- *omvang* $|x|$ van de input x. Die omvang is een of ander natuurlijk getal. Een *tijdcomplexiteit* van een computerprogramma is per definitie: een functie f van $\mathbf{N}$ naar $\mathbf{N}$ (met $\mathbf{N}$ de verzameling van de natuurlijke getallen, inclusief 0) zodanig dat voor elke input x geldt: het aantal rekenstappen van het programma op x bedraagt hooguit $f(|x|)$.

Een programma heet *van polynomiale tijdcomplexiteit* als het van tijdcomplexiteit *f* is voor een of andere polynomiale functie *f*. Er zijn redelijke overwegingen, zowel van praktische als van theoretische aard, om het programma precies in dat geval *efficiënt* (of *efficiënt genoeg*) te noemen, met daarbij de intentie van "snel genoeg in de praktijk". Het gaat hier om een vuistregel die weliswaar niet volmaakt is, maar die over het algemeen behoorlijk voldoet, en wel behoorlijk genoeg.

Achter dit alles zit nog wel de aanname dat de meting van "aantallen rekenstappen" geschiedt aan de hand van een computermodel dat redelijk acceptabel is volgens geëigende criteria. Gebleken is dat de zogenaamde *Turingmachine*, kortweg "TM", geschikt te hanteren is als zo'n model. Zie voor achtergronden bijvoorbeeld: het artikel *Machine models and simulations* van Peter van Emde Boas in het *Handbook of Theoretical Computer Science, volume A* (ed. Jan van Leeuwen).

Een recapitulatie van de TM, en van het corresponderende begrip *TM-programma*, alsmede van verwante begrippen, komt op de presentatie zelf aan de orde, en wel vlot aan de hand van het standaard plaatje (met *band*, *lees-/schrijfkop* en *besturingseenheid*). Doenlijke --- en aardige --- stof ook voor de HBO-doelgroep.

Bij een *deterministisch* TM-programma, kortweg: *DTM-programma*, is telkens hooguit één rekenstap mogelijk. In de algemenere (mogelijk *nondeterministische*) situatie van een TM-programma kan het gebeuren dat er vanuit een gegeven situatie (of "configuratie") van de TM een keuze te maken is tussen verschillende mogelijkheden (zij het nog wel slechts eindig veel mogelijkheden) voor de volgende rekenstap.

De door een TM-programma *M geaccepteerde taal L*(*M*) bestaat per definitie uit alle invoerstrings *w* waarop er overeenkomstig *M* een zodanige berekening (d.w.z.: een rij aansluitend rekenstappen) mogelijk is dat uiteindelijk een zgn. *accepterende toestand* verkregen wordt.

Deel II: hoofdbegrippen uit de theorie van NP-compleetheid.

Allereerst moet hier het begrip "probleem" nader gespecificeerd worden. Een *probleem* (voluit: *beslissingsprobleem*) bestaat uit: (i) een verzameling van zogenaamde *parameters* en (ii) een *vraagstelling* die toe te spitsen is op elk van die parameters en dan telkens hetzij antwoord "ja", hetzij antwoord "nee" heeft. Elk probleem (van "redelijk nette aard") kunnen we identificeren met een formele taal, en wel door de parameters ervan op een natuurlijke manier te representeren door formele expressies. De bedoelde taal bestaat dan precies uit alle representaties van de parameters waarbij antwoord "ja" hoort.

We zijn met name geïnteresseerd in problemen die *efficiënt oplosbaar* zijn; d.w.z.: waarvan de corresponderende taal accepteerbaar is door een DTM-programma van polynomiale tijdcomplexiteit.
Een ruimere (of althans minstens zo ruime) klasse is: de klasse van de problemen die *non-deterministisch efficiënt* oplosbaar zijn. Precieze definitie: vervang direct hierboven "DTM" door TM".

Centraal staan hier de klassen **P** en **NP**. Deze bestaan respectievelijk uit alle problemen die efficiënt oplosbaar zijn, resp. nondeterministisch efficiënt oplosbaar zijn. In de praktijk gaat het uiteindelijk om de klasse **P**, maar de (minstens zo ruime) klasse **NP** blijkt een nuttig of zelfs wezenlijk hulpkader voor nadere studies van **P**. Meer hierover, met voorbeelden ook, tijdens de representatie

NP-complete problemen zijn speciale problemen uit **NP**. Zij hebben, globaal gesteld, de eigenschap dat elk willekeurig probleem uit **NP** ertoe te herleiden is. (In de precieze definitie staat hier in plaats van "ertoe te herleiden is": *polynomiaal ertoe reduceerbaar is*. Dit begrip zal nog nader uitgelegd worden.) De NP-complete problemen zijn dus in zekere zin de moeilijkste problemen uit **NP**. In te zien is: zodra één NP-compleet probleem in **P** zit, zitten alle problemen uit **NP** ook in **P**, hetgeen dan inhoudt: **P = NP**. Vooralsnog is het antwoord op de vraag "**N = NP**?" echter onbekend. Dit is een befaamd open probleem (op een hoger niveau) in de theoretische informatica en wordt ook wel *het P=NP-probleem* genoemd.

Deel III: het bordspel ter verdere verduidelijking van de theorie van NP-compleetheid.

Hoe is (uitgaande van de exacte definitie van "NP-compleetheid") aan te tonen dat er überhaupt NP-complete problemen (van een praktisch redelijk relevante aard) bestaan? En hoe is uitgaande van een of ander NP-compleetheid probleem eventueel ook nog de NP-compleetheid van andere problemen aan te tonen? Dit zijn heel relevante vragen, zowel op theorieniveau als op praktijkniveau.

Tijdens de presentatie wordt op aanschouwelijke wijze een bepaald bordspel geïntroduceerd dat een nader inzicht in (aard en verdere aanpak van) die vragen geeft, en wel op een manier die ook het HBO-publiek kan aanspreken. Aan het bordspel is direkt een bepaald (beslissings)probleem te koppelen, zeg, *het* *bordspelprobleem*, kortweg, *BSP*. In aansluiting op de uitleg van het bordspel zelf zal vlot te beargumenteren zijn dat BSP NP-compleet is. Hier telt, (heel) globaal gezegd, dat overgangen van (correcte vullingen van) rijen naar rijen op het spelbord corresponderen met rekenstap-overgangen tussen configuraties van de TM. (Het bordspel is heel goed op zich te bezien, maar deze verwantschap met TMs is natuurlijk een soort van doorgestoken kaart geweest bij het didactisch gemotiveerde ontwerp ervan.) Verder zal geïllustreerd worden hoe uit de NP-compleetheid van BSP tamelijk voortvarend de NP-compleetheid van diverse andere problemen af te leiden is, inclusief standaard voorbeelden uit de literatuur.

Het idee van een dergelijk didactisch gebruik van een bordspel ter verduidelijking van de wondere wereld van NP-compleetheid is in het verleden onafhankelijk door Peter van Emde Boas (thans: emeritus hoogleraar UvA) en door mijzelf ontdekt. (Schriftelijke neerslag zijnerzijds: het artikel *Dominoes are forever.* Van zijn kant is er ook verwant demonstratiemateriaal. Mijnerzijds was er destijds in dit verband o.a.: RUG-onderwijsmateriaal.) Het idee verdient nog steeds een bredere benutting binnen de ICT-onderwijspraktijk.