

## HOOFDSTUK 8. COMPUTERS

### 8.1 TECHNISCHE ONTWIKKELINGEN

Onder technische ontwikkelingen versta ik de ontwikkelingen met betrekking tot de mogelijkheden die mensen hebben om te manipuleren met energie, materiaal en informatie. Volgens Verhoeff (1980) kunnen in de ontwikkelingsgeschiedenis van menselijke technische voortbrengselen (of "artefacten", een term van Herbert Simon, zie hoofdstuk 13) enkele stadia of niveaus van toenemende complexiteit onderscheiden worden.

1. Het eerste stadium bestaat uit de mogelijkheid om de ruimtelijke structuren te beïnvloeden. Mensen leerden b.v. een huis te bouwen, dat een afscherming bood voor regen en kou. Voordat mensen dat konden, waren ze aangewezen op wat de natuur te bieden had, zoals een hol in een berg. Andere voorbeelden zijn vaatwerk, bruggen. De konstrukties van dit niveau zijn star, en tamelijk grof. De mogelijkheden zijn beperkt door materialenkennis en bewerkingsvaardigheid.
2. Het tweede stadium bestaat uit mogelijkheden om steeds veranderingen in ruimtelijke structuren te kunnen aanbrengen. B.v. scharnierende deuren (beter dan het primitieve rotsblok voor de ingang), het wiel (in zekere zin een vooruitgang t.o.v. de primitieve slede).
3. Het derde stadium geeft mogelijkheden om de krachten die nodig zijn om de bewegingen tot stand te brengen te beheersen. Daartoe moeten energiebronnen in de artefacten worden ondergebracht. Aanvankelijk wordt weer alleen gebruik gemaakt van wat de natuur biedt, b.v. menselijke en dierlijke spierkracht (onder andere in de vorm van tredmolens, galeien) of natuurkrachten, zoals wind en waterstroom. De uitvinding van de stoommachine luidt het tijdperk in dat wel de "eerste industriële revolutie" wordt genoemd. Verbeteringen (vooral qua manipuleerbaarheid) waren de verbrandingsmotor en later de elektromotor.
4. Het vierde stadium geeft de mogelijkheid om de in een artefact ondergebrachte energiebronnen "zichzelf" te laten opstarten, afslaan etc.. Dit realiseren is een kwestie van "regeltechniek". Om een proces te kunnen regelen is informatie nodig, en informatieverwerking om de informatie om te zetten in de juiste signalen voor de motoren. Aanvankelijk konden alleen mensen dit doen, later komen er mechanische

middelen zoals uurwerken met slagwerk. De uitvinding van de computer heeft de mogelijkheid de informatie-inwinning en -verwerking door artefacten te laten uitvoeren aanzienlijk verruimd en verfijnd.

Wat de stoommachine was voor het energiegebruik in artefacten was de computer voor de informatieverwerking, een reden waarom het tijdperk waarin we nu leven wel de "tweede industriële revolutie" wordt genoemd.

De technische ontwikkelingen hangen nauw samen met hoe de wereld er uitziet: de ontwikkelde mogelijkheden vragen alle een eigen infrastructuur. Zo is voor de toepassing van het wiel een wegennet nodig. De mogelijkheden van de stoommachine, een moeilijk manipuleerbare krachtbron, bepaalde de opbouw en organisatie van fabrieken: een grote hal, met daarin een centraal aangedreven as, waaronder machines gegroepeerd werden, aangedreven met riemen. De verbrandingsmotor, die het transport en het toedienen van de energiebron vereenvoudigde, heeft een net van verkooppunten van olieproducten opgeleverd, terwijl grootschalige toepassing van elektromotoren niet goed denkbaar is zonder een elektriciteitsnetwerk.

Computers hebben aanvankelijk eenzelfde centraliserende invloed op de organisatie van het werk gehad als de stoommachine, maar "microcomputers" hebben het fysiek en economisch mogelijk gemaakt om op veel plaatsen ook kleine hoeveelheden informatie automatisch te verwerken.

De technische ontwikkelingen hebben ook grote invloed gehad op het gedrag van mensen. Door de toenemende mogelijkheden van artefacten zijn er voor mensen steeds minder taken overgebleven. Men zegt, dat keizer Vespasianus in de eerste eeuw n.C. sterk gekant was tegen de invoering van het waterrad, omdat hij werkloosheid vreesde. Misschien is het zo, dat de meest recente ontwikkelingen ingrijpender zijn, omdat ze zo snel gaan. Daardoor komt het voor, dat talrijke mensen hun werk zien verdwijnen, zich omscholen en vervolgens ook hun nieuwe taak verliezen, omdat ook daarvoor inmiddels een machine is ontwikkeld.

Volgens Verhoeff gaan de technische ontwikkelingen nog steeds voort. Als vijfde niveau noemt hij de mogelijkheid, dat een artefact "zijn eigen taak wijzigt". Op dit moment kunnen alleen mensen (en dieren) dit nog, b.v. de houthakker die met hakken ophoudt wanneer er een beer aankomt, of als besloten wordt de produktie van iets te verminderen of te beëindigen, wanneer men in de toekomst een afnemende vraag verwacht. Hij noemt dit "een soort strategisch management", dat overigens ook mensen zelf nauwelijks kunnen uitvoeren.

## 8.2 COMPUTERGENERATIES

Computers zijn elektronische dataverwerkende apparatuur. Zij werden

voorafgedaan door niet-elektronische apparaten, zoals de "analytische machine" van Babbage en diverse Hollerith-machines voor verwerking, o.a. sorteren, van ponskaarten.

In de technische ontwikkelingen van computers zijn ook een aantal stadia te onderkennen (vgl. Evans, 1980, blz. 49, Verhoeff, 1979 (1)). De ontwikkeling betreft onder meer de omvang, het energieverbruik, de verwerkingssnelheid, de omvang van de opslagmogelijkheden (gewoonlijk "geheugens" (naar het engelse "memory") genoemd) en de prijs. De stadia die men in de ontwikkeling kan onderscheiden noemt men in dit geval meestal generaties. In Japan wordt thans gewerkt aan het creëren van de "vijfde generatie computers". Het is de bedoeling dat deze computers een zeer grote verwerkingssnelheid zullen hebben en in natuurlijke taal met hun gebruikers kunnen communiceren.

De eerste vier generaties kunnen als volgt gekenschetst worden.

1. Bij de eerste generatie werden eerst een aantal componenten (o.a. talrijke houders voor elektronenbuizen) geplaatst en vervolgens werden deze met elkaar verbonden (bedrading), waarna verder componenten werden toegevoegd. Het bedraden werd al spoedig relatief duur, terwijl deze computers ook veel onderhoud vroegen (vervanging van elektronenbuizen) en veel energie en ruimte vroegen.
2. Bij de tweede generatie legde men eerst de verbindingen, door deze te etsen op kunststof platen, om vervolgens de componenten erin te steken en te bevestigen. Ook een belangrijke component veranderde: elektronenbuizen konden inmiddels vervangen worden door transistoren. Transistoren werden eveneens gemaakt met een soort foto-ets proces, terwijl de elektronenbuizen van de vorige generatie geassembleerd moesten worden uit van te voren apart vervaardigde onderdelen. Door dit procedé konden de componentendichtheid, de snelheid en de betrouwbaarheid sterk toenemen.
3. De derde generatie ontstond toen men erin slaagde de beide drukprocedures van de tweede generatie te combineren. In een proces konden zowel de verbindingen als de componenten vervaardigd worden. Het resultaat daarvan noemde men een "geïntegreerde schakeling" of IC (van "integrated circuit"). Zo'n IC bestaat meestal uit een minuscule schijfje silicium, waarop de componenten en de bedrading zijn gedrukt en een omhulsel, waaraan zo'n acht tot 40 pinnen, die voor verbinding met componenten buiten het circuit zorgen. Zulke IC's kunnen op hun beurt weer onderdeel uitmaken van een tweede generatie circuit, dat weer deel kan uitmaken van een eerste generatie systeem, etc.. Aanvankelijk was het procedé nog weinig verfijnd, en het aantal componenten in een IC nog tamelijk beperkt.
4. De volgende generatie(s) wordt (worden) gevormd door verdere

schaalvergroting van de integratie op een "chip" (een schijfje silicium). Na enige tijd was de techniek ontwikkeld om een complete informatieverwerkingseenheid (of "micro processor") op een chip onder te brengen. Men spreekt van "large scale integration" (LSI) en "very large scale integration" (VLSI).

Voor een computer is niet alleen een verwerkingseenheid nodig maar ook een opslageenheid waarin de data (in de vorm van elektrische stromen) kunnen worden bewaard. Aanvankelijk, in de eerste en tweede generatie, maakte men gebruik van grote aantallen kleine ringmagneetjes, alle voorzien van enkele draden, die wel of niet gemagnetiseerd konden zijn en dus in twee toestanden konden verkeren. Ze konden dus elk een bit (zie hoofdstuk 1) aan informatie bevatten. Later was men in staat ook geheugens op chips onder te brengen. Het is zelfs mogelijk (vgl. Zaks, 1980, blz. 26) een volledige, hoewel beperkte, computer, dus inclusief geheugen op een enkele chip onder te brengen. Op dit moment zijn de toepassingsmogelijkheden van zulke "single chip microcomputers" nog tamelijk beperkt.

Soms wordt gesproken van "microcomputers" ter onderscheiding van "minicomputers" (vgl. Zaks, 1980, blz. 31). Beide vormen een compleet systeem met verwerkings- en opslageenheid, in- en uitvoermogelijkheid en energievoorziening, maar een minicomputer behoort tot de derde, een microcomputer tot de vierde generatie. De ouderwetse omvangrijke computers van de eerste en tweede generatie worden soms "maxicomputers" genoemd. De grenzen vervagen echter, onder meer omdat in minicomputers ook meer en meer grootschalige integratie wordt toegepast.

### 8.3 ENKELE EIGENSCHAPPEN VAN COMPUTERS

Computers bestaan in beginsel uit een aantal relais. Een relais is een elektrische schakelaar die geopend of gesloten kan worden door een elektrische stroom, of de afwezigheid daarvan. Een relais heeft vier aansluitpunten voor elektrische stroom: twee voor de (aangedreven) schakelaar en twee voor de (aandrijvende) stroom. (Er zijn zeer veel soorten relais, met soms veel meer aansluitpunten, maar dat doet hier niet ter zake.) In een computer zijn een aantal van die relais met elkaar verbonden terwijl er tevens is voorzien in een aantal aansluitpennen voor de verbinding naar buiten. Worden twee of meer van die pennen, b.v. bij inschakelen van de computer onder stroom gezet, dan kunnen sommige kontakten daardoor geopend, andere gesloten worden. Dat openen en sluiten leidt, door de interne verbindingen weer tot openen en/of sluiten van andere kontakten, en zo kan men zich voorstellen, dat door het inschakelen van de computer een aantal, en eventueel ook steeds andere, relais gaan staan klepperen. (Dat klepperen zal men meestal niet horen, omdat het systeem niet uit mechanische relais bestaat, maar uit zeer kleine transistoren.) De elektrische spanningen op sommige van de relais kunnen via de aansluitpennen weer

gemeten worden en ook kan men het geklepper trachten te beïnvloeden door op sommige pennen een elektrische stroom, eventueel van korte duur, of in een bepaald patroon aan te sluiten. De relais zijn op een zodanige manier aan elkaar verbonden dat het mogelijk is aan de pennen die daarvoor bedoeld zijn door de ontwerpers, de uitvoerpennen, praktisch ieder gewenst patroon, of reeks van patronen te laten ontstaan, wanneer sommige andere pennen, de invoerpennen, van de juiste patronen worden voorzien. Dat laatste is gemakkelijker gemaakt, doordat die patronen door het stelsel enige tijd kunnen worden bewaard, om vervolgens met de juiste snelheid, na invoer van een bepaald patroon, te worden verwerkt.

Een bepaald patroon, of reeks van patronen, maakt het bovendien mogelijk, om tijdens het verwerken van de invoerpatronen, nieuwe patronen binnen te laten, uit het geheugen (de bewaareenheid), of van buiten.

### 8.3.1 Analoog, Digitaal, Binair

Als een computer zo werkt als hierboven in grote lijnen is aangegeven, moeten de patronen waarmee gewerkt wordt bestaan uit een combinatie van elektrische stroompjes (of spanningen). Als de computer data moet verwerken, dan kunnen die weergegeven worden in de vorm van b.v. nullen (afwezigheid stroom) en enen (aanwezigheid stroom). Die vorm noemen wordt "binair" (tweewaardig) genoemd. De binaire vorm is een bijzonder geval van de "digitale" vorm, die van discrete variabelen (vgl. hoofdstuk 3). Veel signalen hebben een niet-digitale of "analoge" vorm, d.w.z. dat deze informatie zonder verlies van nauwkeurigheid niet in de digitale vorm kan worden omgezet. B.v. de wijzerstand van een meetinstrument, de hoogte van een toon. De begrippen "analoog", "digitaal" en "binair" zijn nauw verwant aan de begrippen "continu", "discreet" en "dichotoom", die ik in hoofdstuk 3 heb behandeld. Meetresultaten hebben de digitale vorm, omdat variabelen altijd discreet gemeten worden. Analoge data zullen voor gebruik in digitale computers eerst "gedigitaliseerd" moeten worden. Als voorbeelden kunnen rasters van foto's (met talrijke zwarte puntjes op een witte ondergrond) en ook geluid kan tegenwoordig met succes in binaire codes worden omgezet. Digitaliseren is soms technisch niet eenvoudig, zoals de "late" uitvinding van de compactdisc leert (het probleem is, dat talrijke malen per seconde, vaker dan de hoogst waarneembare frequentie, het signaal moet worden bemonsterd en het juiste getal vastgelegd), maar heeft wel voordelen. Een voordeel is dus, dat digitale data in computers ingevoerd kunnen worden, maar in het algemeen geldt dat data na digitalisatie gemakkelijker foutloos kunnen worden bewaard of verplaatst. Dit betekent b.v. minder ruis in overgezonden geluid, en minder "sneeuw" op een televisiebeeld.

Data die getallen voorstellen kunnen altijd in binaire vorm, in getallen in een tweetalig stelsel worden omgezet. (Deze methode is echter voor zeer grote en zeer kleine getallen onhandig, omdat het aantal benodigde nullen en enen dan erg groot wordt. Daarom wordt in de praktijk vaak een ande-

re methode gebruikt.) Ook letters, cijfers en andere "tekens" die geen getallen voorstellen kunnen in binaire vorm worden ingevoerd, als voor het gebruik van de benodigde patronen een afspraak is gemaakt. Een veel gebruikte afspraak is de z.g. American Standard Code for Information Interchange of A.S.C.I.I. Dit is een code die voor elke letter van het Anglo-Amerikaanse alfabet, voor de cijfers van het tientallig stelsel en voor sommige andere tekens een patroon geeft, bestaande uit zeven nullen en enen. Het aantal verschillende patronen ("typen") is dus  $2^7 = 128$ . Een rijtje van acht bits noemt men wel een "byte". Een byte is dus voldoende om een teken volgens de ASCII-code te bevatten. Een rij tekens van willekeurige lengte heet tekenrij of in het engels "string".

### 8.3.2 De Opbouw Van Een Computer

Een computer kan omschreven worden als een elektronisch dataverwerkend apparaat dat programma's kan uitvoeren. Een programma is een reeks instructies, waaronder in dit verband de elektrische patronen waarmee aangegeven wordt hoe patronen uit het geheugen moeten worden gemodificeerd worden verstaan. Niet alle patronen stellen een instructie voor. Alle patronen, zowel die welke instructies voorstellen, als die dat niet doen kunnen in het geheugen worden geplaatst, eruit worden gehaald en gemodificeerd.

Iedere computer bestaat (vgl. Zaks, 1980, blz. 20) uit tenminste de volgende drie onderdelen:

1. Een centrale verwerkingseenheid (of CVE, Engels: "central processing unit" of CPU). De CVE kan instructies of andere data uit het geheugen halen, en instructies uitvoeren. (De CVE bevat een eigen - zeer snel - geheugen, "registers" genaamd.)
2. Een geheugen. Een geheugen kan data, waaronder instructies bevatten. Een geheugen (of een deel ervan) kan vluchtig zijn (d.w.z. als de elektrische voeding wegvalt, de data uit het geheugen verdwenen zijn), of niet vluchtig. Geheugens kunnen "beschrijfbaar" zijn en/of "leesbaar". Het ophalen van een patroon uit een geheugen, overigens zonder dat dat patroon daarmee uit het geheugen hoeft te verdwijnen, noemt men "lezen", het plaatsen van een patroon in het geheugen "beschrijven". (Sommige geheugens zijn alleen leesbaar en worden daarom naar het engelse "read only memory" wel "ROM" genoemd.) De capaciteit van een geheugen kan worden uitgedrukt in het aantal bytes dat het kan bevatten. Onder 1 K(ilo)byte wordt  $2^{10} = 1024$  byte verstaan en 1 M(ega)byte is 1024 Kbytes. Iedere byte van een geheugen kan een aanduiding hebben, een "adres". Als een microprocessor voor het weergeven van die adressen b.v. 16 bits gebruikt (2 bytes), dan kunnen daarmee dus  $2^{16} = 65536$  adressen in het geheugen aangeduid worden, corresponderend met een geheugencapaciteit van 64 Kbytes.

3. Invoer- en uitvoermogelijkheden. Deze zorgen voor de verbinding met (apparaten in) de omgeving. Voorbeelden van zulke apparaten zijn een elektronisch toetsenbord, een beeldscherm en een afdrukeenheid (engels: "printer"). Wordt een toets op een toetsenbord ingedrukt, dan gaat er een rijtje (in ASCII zeven stuks) nullen en/of enen naar een onderdeel van de computer. Ontvangt een beeldscherm of een printer een byte, dan kan er een teken op het scherm of papier verschijnen.

Als in- en uitvoermogelijkheid kan men ook een niet-vluchtig geheugen nemen, b.v. een magnetische schijfeenheid. Dat kan dienst doen voor duurzame opslag van data, of als reservegeheugen voor het geval het onder 2. bedoelde geheugen ("werkgeheugen") onvoldoende capaciteit heeft. Een reeks data op zo'n schijfeenheid noemt men wel een "bestand" ("file" in het engels).

Programma's en overige data worden wel met de engelse term "software" aangeduid, de mechanische en elektronische componenten met "hardware". Het onderscheid tussen beide is niet altijd duidelijk, hetgeen ook blijkt uit het feit dat men software bevattend ROM wel "firmware" noemt, een tussenvorm dus.

### 8.3.3 Enkele Soorten Programma's, Computertalen

Het programmeren van een computer door middel van patronen, bestaande uit nullen en enen, werkt niet prettig. Ook na langdurige oefening worden die patronen moeilijk herkend. Daarom heeft men bedacht, dat het prettiger zou zijn, wanneer het programmeren kon plaatsvinden met formules waarin tekenrijen voorkomen die overeenkomen met woorden uit de natuurlijke taal of met afkortingen daarvan. B.v. "ADD R0,R1" zou kunnen betekenen: "tel de inhoud van register R0 op bij die van R1 en plaats het resultaat in R0". Een taal die uit zulke formules bestaat wordt "assembleertaal" genoemd. De verzameling van de patronen van nullen en enen, die de machine "begrijpt", wordt "machinetaal" genoemd. Nu is het dus mogelijk om een programma te vervaardigen in assembleertaal, dat vervolgens om te zetten in machinetaal en het resultaat in te voeren in de computer. Gemakkelijker zou het nog zijn, als men dat de computer zelf kon laten doen middels een speciaal vertaalprogramma. Zulke programma's zijn beschikbaar en worden "assemblers" genoemd. Programma's die het omgekeerde doen (van machinetaal naar assembleertaal) heten "disassemblers".

Ook het programmeren in assembleertaal is nog vrij lastig, o.m. omdat de programmeur allerlei zaken, die de interne werking van de computer betreffen, in de gaten moet houden. Om het mensen nog wat gemakkelijker te maken zijn z.g. "hogere programmeertalen" (hoger dan assembleertaal) ontworpen, zoals Fortran, Cobol, Algol, Basic en Pascal. De uitdrukkingen die in deze talen zijn toegelaten lijken (sterker nog dan as-

sembleertalen) op een zeer eenvoudig Engels, gecombineerd met wiskundige uitdrukkingen. Er zijn ook weer programma's (geschreven in assembleertaal of machinetaal) die programma's in hogere programmeertaal vertalen naar machinetaal (in het engels dikwijls aangeduid als "compilers"). Het programma, geschreven in een hogere programmeertaal noemt men wel "broncode" (naar het engelse "source code"), de vertaling in machinetaal "object code".

Niet zelden zijn verschillende van deze mogelijkheden op een en dezelfde computer aanwezig. Bovendien zitten er aan een computer vaak verschillende in- en uitvoermogelijkheden, soms ook kunnen verschillende gebruikers tegelijk van een computer gebruik maken. Om alle verschillende programma's die op een computers kunnen draaien, met alle in- en uitvoermogelijkheden te kunnen sturen wordt dikwijls een speciaal programma gebruikt. Zo'n programma, dat een computer beschikbaar maakt voor de invoer van programma's en voor de uitvoering daarvan zorgt, wordt aangeduid met "besturingssysteem" ("operating system").

Een algemeen programma, waar veel computersystemen over beschikken, is een z.g. "editor". Dit is een programma waarmee bestanden kunnen worden aangemaakt of veranderd door gebruikers. Een editor maakt van een computer een soort schrijfmachine, alleen de tekst komt niet (rechtstreeks) op papier, maar in het (achtergrond)geheugen. Het wijzigen van teksten, zoals het verbeteren van fouten wordt daardoor zeer eenvoudig. Die teksten kunnen "gewone" teksten zijn, geschreven in natuurlijke taal, maar het kunnen ook programma's zijn, of andere data.

Sommige computertalen (b.v. Basic) danken hun populariteit deels aan het feit dat er een combinatie is ontworpen van compiler, besturingssysteem en editor.

Een laatste bijzonder programma dat ik hier wil noemen is de "bootstrap loader". De naam komt van de Engelse uitdrukking "to lift oneself up by his own bootstraps" Dit "bootstrappen" is een procedé dat kenmerkend is voor de mogelijkheden van computertechniek. Het eerste, minimale programma heeft tot doel een ander, wat omvangrijker programma in het geheugen te halen en op te starten, dat een nog wat krachtiger programma in het geheugen haalt en opstart, etc., totdat uiteindelijk de computer in een vooraf gewenste toestand verkeert. Deze techniek doet enigszins denken aan een schriftelijke cursus lezen (vgl. het in hoofdstuk 3 genoemde "Muenchhauseneffekt").

#### 8.3.4 Computertalen En Natuurlijke Talen

Het is verleidelijk in de vorm van anthropomorfe beelden over dataverwerkende machines te spreken. De gedachte was immers, dat machines dataverwerkende taken van mensen konden overnemen. Van menselijke dataverwerking weet een ieder het een en ander, en de termen die daarbij



gebruikt worden, lijken dan ook handig bij het beschrijven van het gedrag en de mogelijkheden van machines. Het gevaar is dat daarbij aan die machines eigenschappen worden toegedacht die ze niet hebben.

In het bovenstaande is gesproken over "computertalen". De vraag is of die term korrekt is, of het geheel van tekenrijen dat een hogere programmeertaal vormt, inderdaad een "taal" genoemd kan worden, net als natuurlijke talen. Aan de linguïstiek heb ik het onderscheid tussen syntaxis, semantiek en pragmatiek ontleend. Dat onderscheid heeft ook zin bij de hier beschreven computertalen.

De syntaxis van computertalen is eenduidig vastgelegd. Van iedere reeks karakters kan volgens vaste regels worden vastgesteld of ze behoren tot de verzameling "korrekt geformuleerde uitspraken".

Zo kan ook gezegd worden dat aan syntactisch korrekte uitspraken door gebruikers en door computers een betekenis wordt toegekend. Gebruikers hebben in hun hoofd een plan, een concept, hoe de computer gestuurd moet worden. De computer kent aan de uitspraken in de computertaal, de stuurdata, waarin de sturende informatie is neergelegd, ook een "betekenis" toe: de "object code", die door de compiler uit de stuurdata wordt afgeleid.

De object code zorgt er (soms) voor dat het programma ook werkt: de pragmatiek van de computertaal.

Het verschil met natuurlijke talen ligt onder meer in de relatie tussen de syntaxis en de semantiek van de uitspraken. Bij computertalen is die relatie veel vaster dan bij natuurlijke talen. De omzetting van patronen in hogere computertaal naar machinetaal is (vrijwel) gedetermineerd door de compiler. (Vrijwel, want ook compilers kunnen fouten bevatten.) Anderzijds zijn uitspraken in computertalen ook minder redundant dan natuurlijke talen: het weglaten van een teken leidt dikwijls al tot "onbegrip", b.v. blijkend uit een foutmelding door de computer. Uitdrukkingen in computertalen kunnen natuurlijk wel zo geconstrueerd worden, dat ze redundant zijn. Voor het verzenden van gedigitaliseerde signalen kan b.v. extra informatie worden toegevoegd, die het mogelijk maakt fouten te signaleren die tijdens het verzenden zijn ontstaan (z.g. "fouten-ontdekkende codes"). Het is zelfs mogelijk zodanige informatie mee te zenden, dat tussentijds ontstane fouten door het ontvangend systeem worden hersteld ("fouten-herstellende codes").

Bij natuurlijke talen is het, enkele uitzonderingen daargelaten, gewoonlijk niet mogelijk een bepaalde betekenis op een voor een uitleg vatbare manier in de syntaxis neer te leggen. Vaak is het ook niet nodig om dat te doen: een goede verstaander heeft aan een half woord genoeg. De ontbrekende informatie wordt (o.m.) ingevuld, doordat de verstaander kennis van de ervaringswereld heeft.

Bij natuurlijke talen is ook de pragmatiek dikwijls van grote invloed op de betekenisverlening, b.v. bij een stijlfiguur als ironie, terwijl bij het ontwerpen van computertalen juist is geprobeerd zulke onduidelijkheden te vermijden.

Mijn konklusie is, dat computertalen in bepaalde opzichten inderdaad op natuurlijke talen lijken, en dat er dus aanleiding is het gebruik van de term "programmeertaal" te continueren.

### 8.3.5 Programmalussen, Recursie

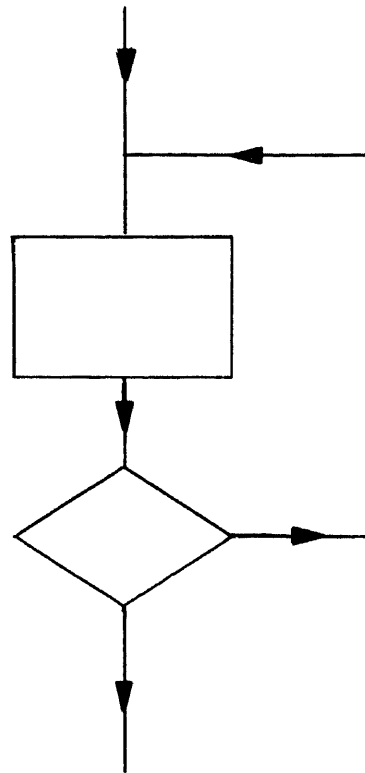
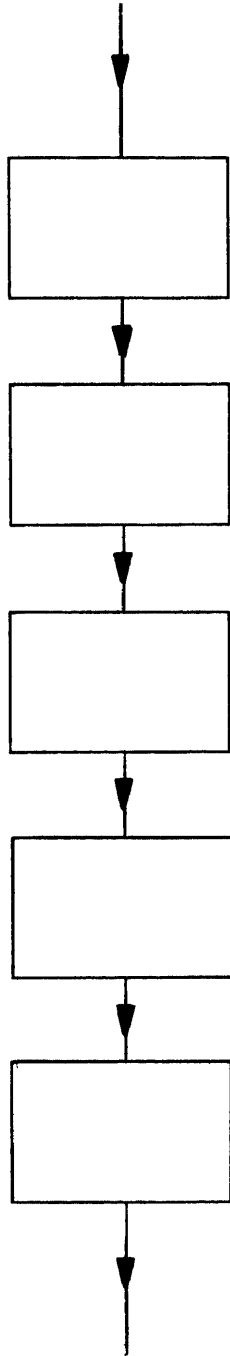
In het algemeen zijn "programmалussen" een belangrijk middel bij het programmeren van computers. Als men een computer iets wil laten doen, is het niet handig alle afzonderlijke handelingen ook alle afzonderlijk te omschrijven. Neem als voorbeeld het sorteren van een groot aantal tekenrijen, b.v. het alfabetisch ordenen van 10000 woorden ("types") uit de Nederlandse taal om een woordenlijst te verkrijgen. Het zou bepaald onhandig zijn een lijst van 10000 instructies in te typen als:

1. "Zoek het alfabetisch eerste woord in de lijst (A) op, schrap het door en zet het in een nieuwe lijst (B)".
2. "Zoek het woord op dat nu alfabetisch het eerste is in lijst A, schrap het door en zet het onderaan de nieuwe lijst (B)".
3. "Zoek het woord op dat alfabetisch het eerste is in de oude lijst (A), schrap het door en zet het onderaan in de nieuwe lijst (B)".
4. Etc., etc.

Al die afzonderlijke instructies lijken nogal op elkaar en het lijkt dus mogelijk om dezelfde instructie maar een keer te geven, en die dan 10000 keer te laten uitvoeren (vgl. fig. 8a):

1. Zoek het alfabetisch eerste woord op in de lijst (A), schrap het door, zet het onderaan in een nieuwe lijst (B), voer instructie 2 uit.
2. Als instructie 1 10000 keer is uitgevoerd dan is het klaar, voer anders instructie 1 uit.

Fig. 8a



Door de overeenkomst in de 10000 handelingen is het mogelijk ze als programmalus te formuleren en de computer doet het werk. De computer moet natuurlijk wel geïnstrueerd worden, wanneer te eindigen, zoals in het eerste deel van instructie 2. Zou de programmeur dit nalaten, dan is in dit geval denkbaar, dat de computer vanzelf stopt, omdat hij in een lege lijst niet kan zoeken. Maar ook is denkbaar, dat hij na 10000 keer de lus te hebben doorlopen terugspringt naar opdracht 1, niets vindt, niets doorschraapt, niets onderaan de nieuwe lijst schrijft, naar opdracht 2 gaat, naar opdracht 1 springt, etc., etc.. Dan is er sprake van wat wel een "eeuwige lus" wordt genoemd.

Binnen een programmalus kunnen weer ander programmalussen optreden. In het gegeven voorbeeld is waarschijnlijk een extra lus nodig om van twee woorden na te gaan welk alfabetisch het eerste is als de eerste letters ervan gelijk zijn. De lus heeft dan tot doel dat naar de tweede, de derde etc. letter gekeken wordt.

Een belangrijke techniek bij het programmeren is "recursie". Om uiteen te zetten wat daaronder verstaan wordt, geef ik eerst een voorbeeld van een definitie waarin van recursie gebruik wordt gemaakt. Hierboven heb ik aangeduid wat onder een tekenrij of "string" wordt verstaan. Een recursieve definitie daarvan zou zijn:

Een "tekenrij" is een karakter of een karakter gevolgd door een tekenrij.

Op het eerste gezicht is deze definitie niet kloppend, omdat in het "definiens" van de definitie het "definiendum" voorkomt, en sommigen zullen beweren dat hiermee niets gedefiniëerd wordt of dat de definitie onbegrijpelijk is. Bij nader inzien blijkt het mogelijk te zijn (als gegeven is wat een "karakter" is en wat niet) aan de hand van de definitie alle "tekenrijen" op te bouwen of uit te maken of iets een "tekenrij" is of niet. Als "k", "a" en "t" karakters zijn is "kat" dan een tekenrij? Toepassing van de definitie leert, dat "kat" een tekenrij is als "at" dat is. "at" is een tekenrij als "t" een tekenrij is. "t" is een karakter, dus volgens de definitie een tekenrij. Dan is "at" en tenslotte "kat" ook een tekenrij. Het is met typografische middelen wat moeilijk voorbeelden te geven van iets, dat geen tekenrij is. In aanmerking komt volgens de definitie: "" (in het Engels wel "null-string" genoemd).

Op vergelijkbare wijze noemt men een recursief programma(onderdeel) wel een programma(onderdeel), dat zichzelf aanroept, dat zichzelf aanroept, dat zichzelf aanroept etc.. Vgl. de Droste-verpleegster, die een doosje draagt, waarop een Droste-verpleegster, die een doosje draagt, etc., of de man (Piet Grijs), die droomde, dat hij droomde, dat hij droomde ....., dat hij wakker werd, dat hij wakker werd, etc, doch, eenmaal uit bed en op straat, zich realiseerde dat hij eenmaal te veel was wakker geworden.

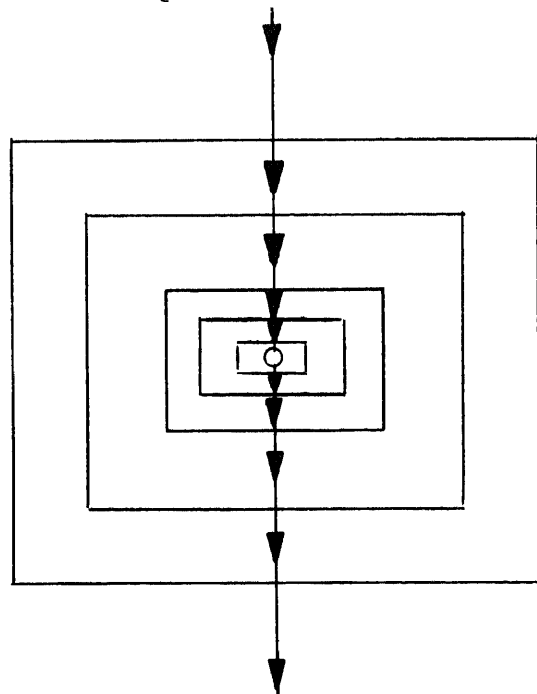
Een alfabetisch geordende woordenlijst zou men kunnen omschrijven als een woord, of een woord gevolgd door een alfabetisch geordende woordenlijst die alleen woorden bevat die alfabetisch na dat woord komen. Met behulp van deze recursieve definitie van een alfabetisch geordende woordenlijst kan nu een programma geformuleerd worden van een ingewikkelder structuur dan het hierboven gegeven, maar dat qua formulering uiterst simpel is:

1. Zoek uit de verzameling woorden de alfabetisch eerste, alfabetiseer de rest en zet het eerste woord er weer voor.

Op het eerste gezicht lijkt het wellicht, dat dit programma onuitvoerbaar is, dat men met zo'n programma niets bereikt, omdat immers het probleem alleen verplaatst is van het alfabetiseren van een verzameling van b.v. 10000 woorden naar een van 9999 woorden. Daarmee is het probleem toch vereenvoudigd, en na 9999 slagen is het probleem teruggebracht tot het triviale probleem van het alfabetiseren van precies een woord. Dan behoeven nog slechts alle woorden, die steeds uit de verzameling zijn gehaald er in omgekeerde volgorde te worden voorgezet.

Het vreemde van de programmalussen is hier, dat het programma zichzelf aanroept, dat er een programma binnen een programma, binnen een programma, binnen een programma etc. is. In een stroomdiagram laat de recursieve structuur zich moeilijk tekenen, anders dan de "gewone" lussenstructuur van de rechter tekening in fig. 8a. Men zou eigenlijk een extra dimensie willen hebben, om niet steeds een kleiner plaatje te hoeven tekenen (vgl. fig. 8b). Sommige programmeertalen, b.v. Lisp, laten een recursieve formulering van het programma, bijna in de vorm van hierboven zonder meer toe. De compiler houdt netjes bij hoe diep de gelaagdheid op ieder moment is, en hoeveel woorden er (in het voorbeeld) aan het eind nog moeten worden voorgezet.

Fig. 8b



Ook in natuurlijke taal komen vreemde lussen voor. Een voorbeeld is de klassieke paradox van de uitspraak van een Kretenser: "Alle Kretensers zijn leugenaars". Ook hier een uitspraak die naar zichzelf verwijst (self-referencing" in het engels). Eenvoudiger gesteld: "Deze uitspraak is niet waar". Deze recursie is niet zinvol, omdat het probleem niet vereenvoudigd wordt door het springen naar een volgende laag.

Een vreemde lus die hier weer sterkt op lijkt bestaat uit twee uitspraken die elkaar ontkennen:

1. Uitspraak 2 is onwaar.
2. Uitspraak 1 is onwaar.

(Juristen zijn met dit soort problemen ook niet onbekend. Art. 11 van de wet Algemene Bepalingen ("De regter moet volgens de wet recht spreken; hij mag in geen geval de innerlijke waarde of billijkheid der wet beoordelen") lokt zulke situaties a.h.w. uit. Bij kwesties van internationaal privaatrechtelijke aard komt het wel voor dat nationale rechtstelsel naar elkaar verwijzen. Rechteren beslissen vaak over hun eigen bevoegdheid, etc..)

Hofstadter (1979, blz. 684 e.v.) meent dat het juist vreemde programmalussen zijn (met veel lagen), die inflexibele machines "intelligent" kunnen maken. Die vreemde programmalussen komen vaak neer op het "springen" van syntaxis van uitspraken (op een zeker niveau) naar de semantiek en/of omgekeerd (naar hetzelfde niveau of een ander). Veel woordgrappen (humor is bij uitstek intelligent gedrag) komen op zulke sprongen neer. B.v. die van het schaap, dat slecht kan lopen, omdat het vroeger lam is geweest.

#### 8.4 MOGELIJKHEDEN MET COMPUTERS

De mogelijkheden van computers liggen op de volgende terreinen:

1. het in- en uitvoeren van data,
2. het verplaatsen van data,
3. het bewaren van data en
4. het bewerken van data.

Tussen deze verschillende mogelijkheden zijn duidelijk onderlinge verbanden. B.v. is het voor het kunnen verplaatsen van data nodig, dat men ze kan bewaren, en is voor het bewaren meestal nodig ze te verplaatsen (naar een bewaarplaats of geheugen).

Het is op dit moment reeds praktisch onmogelijk alle toepassingsmogelijkheden van computers op te noemen, omdat die mogelijkheden zo veelzijdig zijn. Bovendien worden er steeds nieuwe mogelijkheden ontdekt en uitgevonden. Omdat het hier een studie betreft van juridische informatica is het voor sommige toepassingsmogelijkheden minder nodig ze expliciet te vermelden dan andere. Onder de processen die door computers kunnen worden bestuurd kunnen worden onderscheiden (vgl. hoofdstuk 6) data-verwerkende processen en processen waarin geen data worden verwerkt. De laatste zijn voor ons onderwerp minder interessant. Voorbeelden van de eerste zijn er op allerlei terreinen van het maatschappelijk leven:

1. In het industriële productieproces: computers die hele fabrieken of fabrieksonderdelen besturen, van chemische tot koekjesfabriek.
2. In de land- en tuinbouw: computers die landbewerkingsmachines besturen, of de bevochtiging en ventilatie van tuinbouwkassen.
3. In het huishouden: computers die wasmachines of de centrale verwarming besturen.
4. In het onderwijs: computers die vliegtuig- of tanksimulatoren besturen.
5. In het transport: computers die (onderdelen van) auto's, vliegtuigen, treinen, schepen en ruimtevaartuigen besturen.

”Robots” zijn meestal door een computer bestuurd apparaten waarmee voorwerpen kunnen worden gemanipuleerd; zij nemen ook sommige zintuiglijke waarnemingen voor hun rekening.

Er zijn, zoals gezegd, ook talrijke computertoepassingen bij het sturen van dataverwerkende processen. Een tussenvorm zijn processen, waarbij data worden verwerkt, die niet zelf in de sturende computer worden ingevoerd. Men denke b.v. aan een computer die een geautomatiseerde bibliotheek stuurt, waarin alle boeken in bakken zijn geplaatst, die automatisch op rolletjes vervoerd kunnen worden, of een geautomatiseerd archief van microfiches, of een automatisch brievensorteerapparaat bij de posterijen. Bij al die voorbeelden is het onderscheid tussen sturende en gestuurde data dus gemakkelijk te maken.

Bij deze voorbeelden en ook bij de nu nog volgende voorbeelden van computertoepassingen in dataverwerkende processen is het vaak niet uitgesloten, dat ze nu of in de toekomst gebruikt kunnen worden in wetenschap of praktijk van het recht. Alvorens ik nu overga tot een opsomming van de mogelijkheden op juridisch gebied noem ik nog een aantal computertoepassingen in de dataverwerking waarbij toepassing op juridisch gebied voornamelijk niet erg waarschijnlijk lijkt.

Genoemd kunnen worden het gebruik van computers voor het spelen van "arcade-type" spelletjes zoals "pacman" en "space-invaders", maar ook voor simulatie van een of meer spelers en/of attributen in traditionele spelletjes als schaken, bridge, monopolie. Voorts toepassing van computers in het voor recreatieve of andere doelen bewerken van beeld en geluid, b.v. voor tekenfilms, "speciale effecten" in de film, automatisch "inkleuren" van oude zwart-witfilms, "synthesizers", automatische nagalm e.d.. Computers kunnen tegenwoordig ook binnen zekere grenzen door mensen uitgesproken woorden herkennen ("voice input") en het omgekeerde, het in min of meer verstaanbaar geluid omzetten van woorden ("speech synthesizing"). Zeer grote economische belangen vertegenwoordigt het gebruik van computers voor het verwerken van het geldverkeer ("electronic fund transfer") en voor het reserveren van plaatsen in transportmiddelen, vooral vliegtuigen. Hetzelfde geldt voor het gebruik van computers bij het vervaardigen van allerlei industriële ontwerpen, b.v. voor auto's en vliegtuigen. De computer wordt daarbij gebruikt om een ontwerp op allerlei manieren te visualiseren, met perspectivische tekening, doorsneden e.d.. Tenslotte noem ik het gebruik van computers voor allerlei vormen van herkenning van patronen, b.v. met de hand ingevulde geldsbedragen op giroformulieren, het vergelijken van televisiebeelden met elkaar, b.v. ten behoeve van bewaking van een terrein of gebouw, etc..

## 8.5 PROGRAMMATUUR DIE VOOR HET RECHT VAN BELANG KAN ZIJN

In het nu volgende ga ik op bepaalde standaardprogrammatuur, die ik in dit hoofdstuk al heb besproken zoals assemblers en disassemblers, besturingssystemen, compilers en editors niet meer in. (Die reken ik dus vanaf nu tot de apparatuur.)

Het geven van een zinvolle indeling van de programmatuur is een probleem op zich. Veel programma's komen pas tot hun recht, of ontlene hun nut aan hun mogelijkheid tot gebruik in combinatie met andere programma's, zodat de indeling eerder een kwestie is van accenten dan van principes. Als leidraad gebruik ik de indeling in computermogelijkheden die hierboven is genoemd, nl. die van in- en uitvoer, bewaren, verplaatsen en bewerken van data. De geschiedenis van de computertechnologie verschaft een min of meer zinvolle volgorde in de beschrijving.

De eerste computers (in het begin van de jaren vijftig) waren, zoals de naam nu nog verraadt, echte rekenmachines. Geheugens waren duur, omvangrijke achtergrondgeheugens waren nog in het geheel niet beschikbaar. Sommige deskundigen meenden, dat met een aantal computers in de gehele wereld van ongeveer zeven in de gehele behoefte zou kunnen worden voorzien. De eerste computer(s) werd(en) gebruikt om wiskundige tabellen van onder meer logaritmen en goniometrische functies te ver-



vaardigen. Eenmaal met voldoende nauwkeurigheid uitgevoerd, zou dit nooit meer behoeven te gebeuren en de behoefte aan computers zou dus eerder af- dan toenemen. In het begin lag het accent bij computertoepassingen dus vooral op de mogelijkheden tot het verwerken van (kleine hoeveelheden) data.

Al spoedig liet zich de behoefte gevoelen, de te bewerken data en de resultaten ervan in grotere hoeveelheden en voor langere tijd op te kunnen slaan. Met het uitvinden van de mogelijkheid om magneetbanden en magneetschijven als geheugen te gebruiken nemen de mogelijkheden van computers in sterke mate toe. Zij kunnen dan ook gebruikt worden als omvangrijke, snelle en betrouwbare kaartenbakken. Deze geheugens zijn aanvankelijk (in de jaren zestig) nog erg duur, vandaar dat b.v. opslag van data uit boeken in computergeheugens niet plaatsvond door de gehele tekst in een geheugen onder te brengen, maar aldus dat mensen uittreksels maakten die net als bij een kaartenbak b.v. in een trefwoordenregister werden ondergebracht. Voor omvangrijke administratieve toepassingen was deze mogelijkheid in bepaalde gevallen reeds goedkoper dan de handmatige. De in- en uitvoermedia waren nog tamelijk log en de daarvoor gebruikte apparatuur was zwaar, met veel mechanische onderdelen. Veel gebruikt werden ponskaarten en betrekkelijk zware en langzame afdruk-eenheden en elektrische toetsenborden. Sommige onderdelen van de bewerkingsprocessen werden ook nog geheel mechanisch uitgevoerd. Populair waren b.v. machines die ponskaarten konden sorteren.

Met het tegen steeds lagere prijzen beschikbaar komen (in de jaren zeventig) van elektronische toetsenborden voor invoer en beeldschermen voor uitvoer en door de toenemende verwerkingssnelheden wordt het interactieve computergebruik (waarbij de gebruiker continu met de computer kan communiceren) steeds beter mogelijk. Aanvankelijk meestal in de vorm van "time-sharing" (waarbij meer dan een gebruiker tegelijkertijd interactief van een computer gebruik maakt), omdat verwerkings- en geheugeneenheden van voldoende omvang en snelheid nog te duur zijn, maar met de massaproductie en daarmee samenhangende prijsdalingen van verwerkingseenheden en achtergrondgeheugens vindt een verregaande decentralisatie plaats. Kleinschalige administratieve toepassingen (boekhouding, adressenbestanden) en toepassing voor tekstverwerking worden economisch haalbaar, zelfs voor toepassing in zeer kleine bedrijven en huishoudens.

De mogelijkheid van gelijktijdig gebruik van een systeem door verschillende gebruikers had reeds de wenselijkheid geschapen ook op enige, en soms zeer grote afstand van een computer in- en uitvoer te kunnen plegen. Verschillende gebruikers kunnen dan dezelfde data- en/of programma-tuurbestanden aanwenden. Voor bestanden waarvoor het noodzakelijk is dat ze op elk moment voor alle gebruikers hetzelfde bevatten, zoals bij het giraal financieel verkeer of bij reservering van vliegtuigzetels is directe datacommunicatie zelfs noodzakelijk. Via de bestaande telefoonnetten en

later (in Nederland in de jaren tachtig) via speciale elektronische datanetten is het verbinden van in- en uitvoermedia op grote afstand aan centrale computers mogelijk gemaakt. Maar ook bij de toenemende decentralisatie van computergebruik is communicatie op afstand wenselijk. Indien nodig kan men z'n data betrekken van andere kleine computers of van een grote centrale computer, waardoor zeer complexe en steeds wisselende netwerken van computersystemen ontstaan. Deze ontwikkeling is zeker nog niet ten einde. Standaardisatie en het gebruik van nieuwe technieken (b.v. glasvezelkabels in combinatie met lasers) maken nu nog ongekende toepassingen in de toekomst waarschijnlijk mogelijk.

Van de moderne toepassingen is aldus in het algemeen moeilijk te zeggen op welk van de vier aspecten, bewerken, bewaren, in- en uitvoer of verplaatsen, nu de nadruk ligt. De machines paren een grote verwerkingsnelheid aan omvangrijke geheugens, terwijl het gebruik interactief is. Voorts kan de configuratie zodanig zijn dat wanneer bepaalde informatie of bepaalde andere faciliteiten niet ter plekke aanwezig zijn, contact met systemen elders gemakkelijk gerealiseerd kan worden. Bij sommige (vooral de wat oudere), concrete toepassingen is het veelal nog wel mogelijk te zeggen op welk aspect nu de nadruk ligt. Vandaar de nu volgende indeling van - in wetenschap en praktijk van het recht bruikbare - toepassingsmogelijkheden van elektronische dataverwerkende apparatuur.

#### 8.5.1 Accent Op Het Bewerken Van Data

Onder deze categorie kunnen in de eerste plaats gerekend worden programma's voor het uitvoeren van allerlei rekenkundige en statistische bewerkingen op data (die daarbij gewoonlijk als getallen geïnterpreteerd worden).

##### 8.5.1.1 Elektronische kladblokken

De hedendaagse pendant van de allereerste computerprogramma's die wiskundige tabellen vervaardigden zijn programma's die als "elektronisch kladblok" ("electronic spreadsheet") fungeren.

De gebruiker beschikt daarbij over een matrix van geheugenplaatsen waarin getallen kunnen worden opgeslagen. Voor iedere geheugenplaats kan een formule worden opgegeven, volgens welke de geheugenplaats met een getal gevuld moet worden. In de formule kan gebruik gemaakt worden van de inhoud van andere geheugenplaatsen. Ook de formules behorend bij ieder "hokje" wordt bewaard. Het scherm fungeert als een "raam" op de getallenmatrix.

Dit soort programma's is natuurlijk ook geschikt voor het maken van statistische berekeningen, maar de gebruiker moet daarvoor de toe te passen formules en procedures zelf aangeven.

### 8.5.1.2 Statistische pakketten

''Statistische pakketten'' zijn computerprogramma's (of een geordende verzameling daarvan) die een groot aantal statistische formules en procedures kunnen uitvoeren. Een bekend voorbeeld is het ''S.P.S.S'' (Statistical Package for the Social Sciences). Dit instrument is vooral bedoeld voor beoefenaren van gedragswetenschappen en er liggen de volgende doelstellingen aan ten grondslag:

1. De in het programma opgenomen statistische procedures moeten wetenschappelijk en statistisch juist zijn.
2. De programma's moeten in ontwerp en uitvoering tot efficiënt computergebruik leiden.
3. De logica en syntaxis van het systeem moeten overeenkomen met de manier waarop beoefenaren van gedragswetenschappen dataverwerking benaderen.
4. Het systeem bevat statistische procedures (en faciliteiten voor het manipuleren van gegevens i.h.a.), toegesneden op de specifieke behoeften van beoefenaren van empirische gedragswetenschappen.

Statistische pakketten zijn buitengewoon populair en worden op grote schaal gebruikt. Ook in commercieel opzicht zijn ze een succes.

Om in de specifieke behoeften van de gebruikers te kunnen voorzien is het aantal in de pakketten opgenomen procedures in de loop der tijd zeer groot geworden. Zodra er een nieuwe statistische techniek wordt uitgevonden wordt deze aan de pakketten toegevoegd. Ook in ander opzicht vertonen dit soort pakketten de neiging tot uitdijen. In de loop der tijd zijn ze uitgebreid met allerlei voorzieningen, die niet alleen voor statistische toepassingen van belang zijn. Sommige pakketten hebben zeer uitgebreide faciliteiten voor het bewaren en manipuleren van gegevens (zie hieronder) en worden door sommige gebruikers voornamelijk daarvoor gebruikt. Ook zijn er inmiddels interactieve versies en voor kleine computers geschikt gemaakte versies beschikbaar.

Aan het gebruik van bedoelde pakketten zijn een aantal bezwaren verbonden. O.m. vanwege het commerciële karakter zijn de programma's en routines die onderdeel van de pakketten uitmaken niet voor de (gewone) gebruikers toegankelijk. Die gebruikers kunnen niet nagaan hoe de berekeningen intern nu precies plaatsvinden. Dit is geen onbelangrijk bezwaar gezien de eis van duidelijkheid van de methode die gevolgd is bij de productie van uitspraken, gedaan in het kader van wetenschapsbeoefening. Wanneer fouten optreden, kunnen deze het gevolg zijn van fouten in de ingevoerde data en opdrachten, maar regelmatig worden ook fouten in de programma's zelf aangetroffen. Soms hebben die fouten er reeds jaren in-

gezet, eenvoudig omdat de gebruikers aannamen dat de procedures wiskundig en statistisch juist waren. In veel gevallen kan de vraag gesteld worden in hoeverre zulke pakketten nu inderdaad tot efficiënt computergebruik leiden. Ook voor het uitvoeren van triviale berekeningen op kleine dataverzamelingen moet een groot deel van het arsenaal aanwezig zijn en soms zelfs in het werkgeheugen gebracht worden. De uitvoer per printer is gewoonlijk fraai verzorgd, maar uiterst uitvoerig: als hij geen gebruik maakt van de mogelijkheid om beperkingen aan te brengen krijgt de gebruiker vaak aanzienlijk meer berekeningsresultaten dan hij nodig heeft.

Een van de alternatieven waarover gebruikers beschikken is zelf op hun speciale behoeften afgestemde, relatief kleine programma's te vervaardigen. Daarvoor dienen zij precies te weten welke berekeningen voor een bepaalde statistische procedure in welke volgorde dienen te worden uitgevoerd. Naar mijn mening komt deze werkwijze daarom het korrekst interpreteren van de resultaten ten goede. Ook foutendetektie wordt gemakkelijker. Hogere programmeertalen komen deze gebruikers in bepaalde opzichten meer en meer tegemoet, door het opnemen van meer complexe wiskundige bewerkingen (zoals b.v. bepaalde matrixbewerkingen) en het vergemakkelijken van het opmaken van de uitvoer.

Deze aanpak is noodzakelijk, wanneer men modellen gebruikt die niet behoren tot het op een bepaald moment standaard statistisch arsenaal, of wanneer men modellen wil toepassen voor het trekken van konklusies waarvoor de procedures in de standaardprogrammatuur niet voorzien zijn. Als voorbeeld van dit laatste kunnen sommige onderdelen van het in hoofdstuk 11 beschreven project "analyse van rechterlijke beslissingen" genoemd worden.

#### 8.5.1.3 Computersimulatie

Er zijn verschillende mogelijkheden voor een onderzoeker om de effecten van wijzigingen in een (concreet) systeem en/of van de invoer in een systeem te weten te komen. Een mogelijkheid die in de natuurwetenschappen en tot op zekere hoogte in de psychologie aanwezig is, is die van het experiment. Hieronder wordt in dit verband verstaan een onderzoekopzet, waarbij alle relevant geachte variabelen onder strikte controle staan van de onderzoeker, althans op ieder moment bekend zijn. In de maatschappijwetenschappen kan zo'n situatie slechts hoogst zelden gecreëerd worden. Soms doen zich vanzelf situaties voor, waarin alle belangrijk geachte variabelen bekend zijn en/of beheerst kunnen worden, zodanig dat bepaalde effecten gemeten kunnen worden. Is dit tot op zekere hoogte het geval, dan wordt soms ook wel van een "quasi-" of "pre-experimentele" opzet gesproken (vgl. Adams, 1975, blz. 60 en Campbell, 1975, blz. 297).

Is het element van beheersing, althans bekendheid van de van belang geachte variabelen gering, dan kan men altijd nog trachten over een bepaalde periode zoveel mogelijk data te verzamelen over de van belang geachte variabelen en vervolgens trachten met behulp van een bepaald statistisch model tot voorspellingen in het geval van wijzigingen in de variabelen te komen. Niet zelden schiet deze werkwijze te kort, b.v. door de niet beschikbaarheid of te grote kostbaarheid van bepaalde data, of doordat de data als ze al beschikbaar waren uitsluitend betrekking hebben op toestanden van het onderzochte systeem waarin de onderzoeker nu juist niet geïnteresseerd is. Ook kunnen alle beschikbare statistische modellen te kort schieten, doordat ze onvoldoende op de situatie zijn toegesneden (niet isomorf zijn met het onderzochte systeem) en/of onvoldoende nauwkeurige resultaten geven.

In sommige gevallen kan een computersimulatie, al dan niet in aanvulling op andere wijzen van aanpak, een antwoord geven op de vraag, wat er bij wijziging van bepaalde variabelen in een bepaald systeem zou gebeuren.

Bij computersimulaties laat men een computer een deel van de werkelijkheid nadoen. Daarvoor dient men te beschikken over een (formeel) model van dat deel van de ervaringswereld, en men kan de computer wanneer het model daarin is ingebracht gebruiken om een of meer variabelen zeer frequent te wijzigen en bij te houden welke waarden de andere variabelen als gevolg van deze wijzigingen aannemen. Uit de patronen, die aldus gevonden worden kan men soms hetzelfde of meer afleiden dan met een vollediger of complexer model mogelijk was geweest. Als simpel voorbeeld kan het bepalen van het oppervlak van een cirkel in een bepaalde concrete situatie, b.v. voor het bepalen van het benodigde afdek materiaal in een tuin, genomen worden. Men kan dan een computer willekeurig (of volgens een raster) een groot aantal punten laten kiezen in een vlak, waarin de cirkel is voorgesteld. De verhouding van het aantal punten binnen de cirkel en het totaal aantal punten is een indicatie voor het oppervlak van de cirkel, gegeven het oppervlak van het buitenvlak. Bij een cirkel is zo'n oppervlaktebepaling nogal overbodig, omdat de formule daarvoor algemeen bekend is, maar er zijn figuren waarvoor deze methode handig kan zijn.

Er zijn zeer vele computerprogramma's vervaardigd voor het simuleren van allerlei systemen (vgl. Byte, vol. 9 no. 3 (maart 1984), blz. 93 e.v.). Ook in het recht, waarbij men kan denken aan problemen m.b.t. wachttijden bij gerechtshoven, case load studies in het reclassingswerk of bij advocaten (vgl. Salzman, 1974, Smith, 1975, Adams, 1975, blz. 93).

Ook bij simulatieprogramma's kan men weer kiezen tussen "kant en klare" omvangrijke pakketten, of het zelf vervaardigen van een op de situatie toegesneden programma. Ten behoeve van dat laatste zijn zelfs speciale hogere programmeertalen vervaardigd, waarvan Simula er een is.

#### 8.5.1.4 Vertalen per computer?

Zeer tot de verbeelding van sommigen spreekt de gedachte, dat een computer gebruikt zou kunnen worden om geschriften van de ene natuurlijke taal in een andere te vertalen. Brandt-Corstius (1978, blz. 20) vermeldt sommige van de talrijke pogingen die zijn gedaan sinds Weaver in 1949 de mogelijkheid opperde.

In beginsel lijkt de zaak simpel, zeker als twee talen dezelfde lettertekens gebruiken. Gegeven zijn dan twee verzamelingen lettertekens; een geschrift in de ene taal en de vertaling daarvan in de andere. Gevraagd is de verzameling regels volgens welke de ene verzameling uit de andere "berekend" kan worden. Om niet voor ieder tweetal talen opnieuw alle regels te hoeven vinden lijkt het een goed idee om een "tussentaal" te gebruiken (door Brandt-Corstius het "Machinees" genoemd). Het aantal benodigde regelverzamelingen is dan gelijk aan slechts het aantal talen waartussen men wil kunnen vertalen. Op het eerste gezicht behoeven de regels waaruit de computerprogramma's worden samengesteld ook niet al te ingewikkeld te zijn: met "woord voor woord" vertaling komt men al een heel eind. B.v. "De kat is in de tuin" wordt volgens een simpel algoritme "The cat is in the garden". (Als men het Engels of een andere natuurlijke taal als tussentaal gebruikt is het aantal benodigde regelverzamelingen nog een kleiner.)

In de praktijk blijken de pogingen op grote moeilijkheden te stuiten. Door de Amerikaanse overheid gesubsidieerde pogingen werden in 1966 radicaal stopgezet, omdat na een investering van 20 miljoen dollar het vertalen van een "algemene wetenschappelijke tekst" (of is het "algemeen wetenschappelijke tekst": "general scientific text"?) per computer niet mogelijk is en ook geen direkt perspectief heeft (Brandt-Corstius, 1978, blz. 22).

Omstreeks 1980 wordt van de E.E.G. door deelnemers aan het zgn. "Eurotra" projekt een bedrag van ongeveer 25 miljoen gulden gevraagd om in vijf jaar tijd een automatisch vertaalsysteem tussen de zes gemeenschapstalen tot stand te brengen. Japanse experts menen tot in de negentiger jaren werk te hebben bij het ontwikkelen van hun "vijfde generatie computers", maar die moeten meer kunnen dan alleen natuurlijke talen begrijpen.

Bar-Hillel (1964) wijst op sommige van de wellicht onoverkomelijke problemen. Ik citeer Brandt-Corstius (1978, blz. 22): "zijn argumenten komen er in het kort op neer dat voor formeel vertalen behalve een woordenboek ook een encyclopedie nodig is, en wel een encyclopedie waarin alle kennis over de wereld is te vinden. In de zin "Ik kocht een piano en een viool die ik onder de arm mee naar huis nam" moet je bij vertaling in bepaalde talen weten of dat "die" op "piano" en "viool", of alleen op "viool" slaat. Bij elk substantief moet dus vermeld zijn of het onder-de-

arm-mee-naar-huis-draagbaar is." Met zo'n - niet bestaande - encyclopedie zouden sommige semantische problemen bij het vertalen opgelost kunnen worden. Maar er zijn ook problemen die de pragmatiek betreffen. In een kolderverhaal kan iemand heel wel een piano onder de arm mee naar huis nemen, en in sommige gevallen wil de schrijver wellicht nadrukkelijk in het midden laten of een voorwerp, dat syntactisch en semantisch gesproken, mee naar huis genomen zou kunnen worden, in het verhaal ook daadwerkelijk verplaatst wordt. Bij het interpreteren van juridische teksten doet dit soort problemen de pragmatiek betreffende zich niet zelden voor.

Naar mijn mening is het ook wat teveel gevraagd om de computer nu maar meteen te willen programmeren (zoals in het Eurotra-project) voor het interpreteren van nuances zoals die in juridische geschriften voorkomen. Er zijn computerprogramma's die binnen een zeer beperkt kader eenvoudige en zinvolle uitspraken kunnen genereren en interpreteren. Een voorbeeld hiervan is Winograd's programma "Shrdlu" (Winograd, 1972). Sommige problemen van het complexe verband tussen syntaxis en semantiek kunnen in dit zeer beperkte kader (een "wereldje" van een aantal blokken) blijkbaar opgelost worden (vgl. Hofstadter, 1980, blz. 627 e.v.). Het is naar mijn mening een open vraag of en wanneer deze pogingen op het terrein van de "kunstmatige intelligentie", waarover later meer, tot heuse vertaalmachines zullen leiden. Daarbij is het niet uitgesloten, dat juist juridische geschriften, voorzover daarin, zoals nu dikwijls het geval is, van onduidelijke begrippen gebruik wordt gemaakt, voor machine-intelligentie uiterst moeilijk te interpreteren en te vertalen zullen zijn.

Een recente ontwikkeling in de toepassing van computers bij het vertalen van geschriften is er een waarbij een interactief computerprogramma aan een vertaler steeds van ieder woord en/of zinsdeel een aantal alternatieve vertalingen presenteert. De vertaler kiest daar dan uit, of typt een eigen vertaling in wanneer naar zijn oordeel geen van de vertalingen voldoet. Het is niet uitgesloten dat zo'n toepassing van computers, zeker wanneer de oorspronkelijke tekst toch al in voor een computer leesbare vorm beschikbaar is, tot een kostenverlaging van het vertalen leidt.

## 8.5.2 Accent Op Bewaren Van Data

### 8.5.2.1 Programma's voor bestandsbeheer

Voor grote en kleine computersystemen zijn programmapakketten beschikbaar die tot doel hebben het werken met data in de vorm van bestanden te ondersteunen. (De Engelse term voor zulke pakketten is "database management systems". Het gehele systeem van zo'n pakket tesamen met de computer waar het is ingevoerd alsmede de bestanden noemt men wel een "databank".) Ze zijn meestal ontworpen voor het besturen van grote en dikwijls complexe bestanden. Ze bieden (vgl. Lindemann, 1974 blz. 415) in het algemeen de volgende faciliteiten:

1. het opbouwen van bestanden
2. het samenvoegen van bestanden
3. het verwijderen, toevoegen of wijzigen van onderdelen van bestanden
4. het veranderen van de structuur van de data-opslag, hetzij van de samenstelling van de onderdelen, hetzij van de structuur van het bestand als geheel
5. het selekteren van data, o.m. ten behoeve van het genereren van overzichten van de data uit de bestanden of het maken van uittreksels
6. het sorteren van data uit de bestanden, overzichten of uittreksels
7. het beveiligen van de data tegen ongewenste uitvoer of ongewenste wijzigingen.

Vooraf bij omvangrijke bestanden kunnen de wachttijden voor de gebruiker bij het uitvoeren van de bovenomschreven taken vrij lang worden. De mate waarin de ontwerpers van het pakket erin geslaagd zijn deze tijden kort te houden is dan ook een belangrijk beoordelingscriterium voor deze programma's voor bestandsbeheer. Soms worden in de apparatuur voorzieningen ingebouwd om de toegangstijden kort te houden, b.v. halfgeleidergeheugens die dienst doen als tijdelijke opslag voor veelgebruikte onderdelen van een bestand. Andere beoordelingscriteria zijn, naast uiteraard de juiste werking o.a. de efficiency bij het gebruik van de geheugenruimte en de flexibiliteit voor de gebruiker bij het kiezen van de datastructuur en de mate waarin verbanden kunnen worden gelegd tussen onderdelen van de bestanden.

Voor grote computersystemen zijn er talrijke pakketten beschikbaar, vaak vervaardigd door of in opdracht van de fabrikant van de apparatuur. Ook voor microcomputers zijn er verschillende programma's voor het manipuleren van bestanden beschikbaar, waarvan sommige zeer veelzijdig zijn. Net als bij de hierboven behandelde statistische pakketten kan het voor gebruikers voordelen hebben speciale programma's te (laten) schrijven voor de gewenste toepassingen.

#### 8.5.2.2 Geautomatiseerde documentatie

Een bijzonder geval van de hierboven beschreven mogelijkheden vormt het gebruik van computers voor documentatie. Onder documentatie wordt hier verstaan het gehele proces van verzamelen, selekteren, opbergen, terughalen en verspreiden van data.

Zolang er al media zijn waarop data kunnen worden vastgelegd is er al geweest op het grote aantal en de onoverzichtelijkheid van geproduceerde



en/of op schrift gestelde uitspraken op een bepaald terrein (zie b.v. Hugo van St. Victor, in: Migne, *Patrologia Latina*, deel 176, blz. 803-805). Voor dit probleem worden verschillende oplossingen geopperd. Men kan b.v. denken aan het beperken van het aantal uitspraken en/of het aantal geschriften. Een andere oplossing is het beperken van terreinen: "specialisatie", een vorm van arbeidsverdeling (vgl. Bar-Hillel, 1963, blz. 365 e.v.). Met de uitvinding van computers komt er een derde mogelijkheid in het verschiet: automatisering van (een deel van) het documentatieproces.

Aanvankelijk (in de jaren zestig) toen computergeheugens nog uitermate duur waren, was het nog niet haalbaar alle teksten volledig in gedigitaliseerde vorm op te slaan. De automatisering van de documentatie was beperkt tot computerverwerking van sturende data voor het vinden de gewenste documenten: de geautomatiseerde bibliotheekcatalogus. (Ik zie hier even voorbij aan andere vormen van automatisering, die niet rechtstreeks te maken hebben met het opslaan van de inhoud van documenten in computers, zoals geautomatiseerde boekenmagazijnen of magazijnen voor microfiches.)

Een "tussenfase" treedt in, wanneer men het bezwaar van de geautomatiseerde catalogus, nl. dat de data zelf niet per computer geraadpleegd kunnen worden, tracht te ondervangen door van ieder boek, tijdschriftartikel e.d. tevens een samenvatting in het computergeheugen op te nemen. Die samenvatting kan b.v. bestaan uit enkele sleutelwoorden en dan is het verschil met de geautomatiseerde catalogus nog niet zo groot. Gebruikers zoeken dan naar hun gewenste data door het opgeven van een combinatie van toepasselijke sleutelwoorden. De samenvatting kan ook uitvoeriger zijn, waardoor enerzijds voor de gebruiker het zoeken wat moeilijker wordt maar hij krijgt daarvoor een grotere hoeveelheid rechtstreeks opvraagbare data. (In hoeverre de gevonden data ook antwoord geven op de vragen die de gebruiker heeft, m.a.w. in hoeverre ze informatie bevatten, is een vraag die hieronder behandeld wordt.) De samenvattingen kunnen gemaakt worden door speciaal opgeleide documentalisten. Dat maakt ze duur, terwijl ze fouten kunnen bevatten en in ieder geval tijdgebonden zijn. (Er is ook wel geopperd de samenvattingen per computer te laten vervaardigen, maar de problemen die daarbij optreden zijn deels dezelfde als die bij het automatisch vertalen en deels nog moeilijker.)

In de jaren zeventig zijn de opslagmogelijkheden van computers zodanig omvangrijk en goedkoop geworden dat systemen waarbij de volledige tekst van de documentatie in de computer wordt opgeslagen haalbaar worden ("full text" systemen). Gebruikers kunnen met speciale commando's de gehele tekst doorzoeken en ieder woord van de tekst is a.h.w. sleutelwoord.

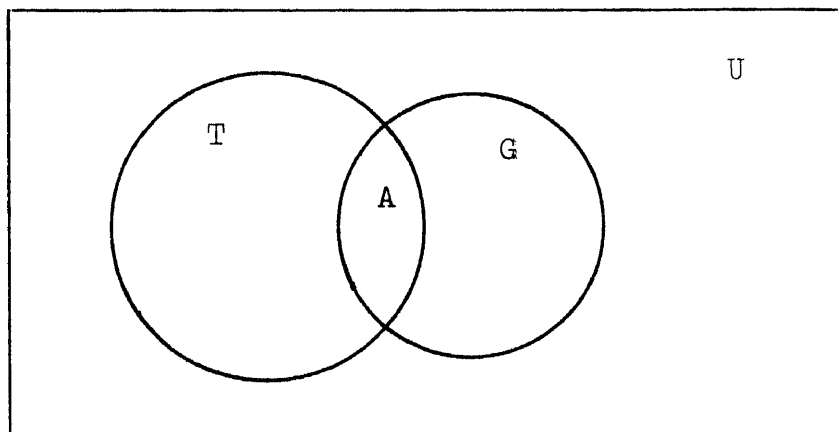
Op juridisch terrein is thans in Nederland een groot deel van de documentatie in gedigitaliseerde vorm opvraagbaar. Bij dit soort documentatie gaat het gewoonlijk om grote databestanden, zodat vooralsnog weinig de-

centralisatie plaatsvindt. De firma Kluwer exploiteert een jurisprudentie-databank en een databank (in de vorm van samenvattingen) voor overige literatuur (waaronder "dogmatiek"). Een databank voor wetgeving kan men raadplegen bij uitgeverij Vermande. Voor ontwikkelingen buiten Nederland, prijzen e.d. zij verwezen naar Eilders, 1983 en Franken, 1981.

De programmatuur die gebruikt wordt voor volledige tekstdocumentatie is natuurlijk niet alleen van belang voor juridische toepassingen. Er zijn kant en klare databankpakketten beschikbaar, soms weer gebonden aan toepassing op systemen van een bepaalde fabrikant. Er zijn ook sommige speciaal voor juridische documentatie geschikte pakketten ontwikkeld, die echter ook voor andere toepassingen handig zouden kunnen blijken.

Ik wil deze paragraaf besluiten met enkele opmerkingen over het beoordelen van sommige kwaliteitsaspecten van de hier bedoelde systemen. Veronderstel dat de documentatie is opgesplitst in documenten (b.v. afzonderlijke rechterlijke beslissingen in een jurisprudentiebestand, of wetsartikelen in een bestand van wetgeving). Men is dan op zoek naar alle documenten over een bepaald onderwerp en niet naar documenten over andere onderwerpen. Deze dubbele dichotomie (wel/niet gevonden resp. wel/niet gewenst) kan worden weergegeven in een z.g. Venn-diagram (zie fig. 8c).

Fig. 8c



Er is een universum  $U$  (alle documenten in het bestand) en een verzameling gewenste documenten  $G$ . Veronderstel dat het zoeken in het bestand sommige van de gewenste documenten alsmede een aantal niet gewenste oplevert. Noem de verzameling gevonden documenten  $T$ . Veronderstel dat de verzameling documenten die bestaat uit de doorsnijding van  $T$  en  $G$  (het gebied waar ze elkaar overlappen)  $A$  wordt genoemd.  $A$  bestaat dus uit alle documenten die en gewenst en teruggevonden zijn. Neem tevens aan dat  $G$ ,  $T$  en  $A$  resp.  $g$ ,  $t$  en  $a$  documenten bevatten.

Onder de "terughaalfactor" ("recall") van deze zoeksituatie wordt verstaan de verhouding tussen a en g (a/g).

Onder de "precisie" ("precision") van deze zoeksituatie wordt verstaan de verhouding tussen a en t (a/t).

Idealiter geldt: terughaalfactor = precisie = 1. De verzamelingen T en G vallen dan geheel over elkaar. De laagste waarde van terughaalfactor en precisie is 0. G en T overlappen elkaar dan in het geheel niet.

In de praktijk is de precisie gewoonlijk gemakkelijker te bepalen dan de terughaalfactor. Voor de precisie hoeft men slechts te weten welk deel van de teruggevonden documenten bruikbaar is. Voor de terughaalfactor moet men weten hoeveel relevante documenten er in het universum waren. Meestal weet men dat niet, en als men het weet hoeft men waarschijnlijk niet meer te zoeken. In het projekt dat in hoofdstuk 11 beschreven wordt, werd voor het selekteren van geschikte arresten van Kluwers databank gebruik gemaakt. Een terughaalfactor van grofweg 50% bleek reeds moeilijk haalbaar te zijn. De hier behandelde criteria zeggen bij het gebruik van volledige tekstsysteem in zekere zin echter meer over de gebruiker dan over het systeem. Niettemin kan de vraag gesteld worden, of de semantische problemen bij het zoeken in zulke systemen de technische mogelijkheden niet in belangrijke mate teniet doen. Anders gezegd, de term "information retrieval system" is voor de hier bedoelde systemen te pretentius: zij geven geen antwoorden op vragen van de gebruiker, maar geven slechts een selectie van data uit de verzameling. (Met name in frans-talige gebieden in Europa wordt wel beweerd dat z.g. "thesauri", gestructureerde woordenlijsten, een oplossing kunnen bieden voor deze problemen. Zie b.v. Bauer-Bernet, 1980.)

### 8.5.3 Accent Op In- En Uitvoer Van Data

#### 8.5.3.1 Administratieve toepassingen

In diverse administratieve toepassingen ligt de nadruk op het in- en uitvoeren van data, al zijn de andere aspecten en met name het bewaren uiteraard ook van belang. De hier bedoelde toepassingen kunnen zowel betrekking hebben op zeer omvangrijke dataverzamelingen, zoals het gebruik van computers voor het bijhouden van de saldo's van alle rekeninghouders door de Postgiro en voor het bijhouden van omvangrijke registers, zoals bevolkingsregisters en registers voor de kamers van koophandel, als op relatief kleine bestanden, zoals bij de boekhouding van kleine bedrijven of huishoudens, of kleine registers zoals sommige persoons- en adressenbestanden. Computers verschaffen tegenwoordig goedkoop permanente in- en uitvoermogelijkheden en voor allerlei administratieve toepassingen is dat handig. Behalve aan bovenstaande voorbeelden kan gedacht worden aan programma's voor het toegankelijk maken en wijzigen van personeels-, leden- en klantenbestanden en aan bestanden voor financiële, goederen- en dienstenadministraties.

### 8.5.3.2 Programma's voor tekstverwerking

Op een schrijfmachine voert men een tekst in die tegelijkertijd weer wordt uitgevoerd. Maakt men typfouten dan moeten die op de uitvoer worden hersteld of een groot deel van de invoer moet worden overgedaan. Bij gebruik van een computer kan men de ingevoerde tekst enige tijd bewaren in het geheugen en alle gewenste wijzigingen aanbrengen voordat de uitvoer op papier plaatsvindt. Bij toepassing van een computer voor dit doel komen er tegelijk allerlei andere mogelijkheden binnen bereik. Men kan b.v. dezelfde brief aan een aantal verschillende geadresseerden willen verzenden en iedere brief z'n eigen opschrift geven. Ook kan men in de brief enkele passages toevoegen, wijzigen of weglaten. Men kan een brief, of een akte of een vonnis, ook geheel samenstellen uit standaardpassages, die in een achtergrondgeheugen zijn opgeslagen.

Computers kunnen de tekst links en rechts netjes "uitlijnen", zorgen dat ingesprongen wordt bij nieuwe alinea's, zorgen voor de juiste paginering, hoofdstukken en paragrafenindeling en zelfs automatisch inhoudsopgaven en indexen maken.

Bij de wat oudere programma's vindt dit soort tekstverwerking dikwijls in twee fasen plaats: de tekst wordt ingevoerd en/of gewijzigd met behulp van een eerste programma (een editor, al dan niet voorzien van of gekoppeld aan een mogelijkheid tot bestandsmanipulatie), waarbij tevens typografische commando's kunnen worden ingevoerd, zoals "begin een nieuwe regel" of "einde paragraaf". De tweede verwerkingslag bestaat uit het uitvoeren van een tweede programma, dat uit het aldus gevormde "bronbestand" de uitgelijnde e.d. tekst in bestandsvorm maakt. Die tekst kan dan vervolgens afgedrukt worden. Sommige nieuwere programma's voeren de typografische commando's a.h.w. direkt uit en de bewerkte tekst kan in de eindvorm direkt op het scherm verschijnen.

Het behoort ook tot de mogelijkheden de computer een korrektie van de spelling uit te laten voeren, doordat in het bestand een woordenlijst is opgenomen. Bij woorden die niet in die lijst voorkomen geeft de computer dan b.v. een teken, en de gebruiker kan zijn fout dan herstellen, of het woord ook in de woordenlijst laten opnemen. Er zijn - ook voor het Nederlands - programma's gemaakt die woorden bijna foutloos kunnen afkorten, hetgeen van pas komt bij het uitlijnen van teksten b.v. voor krantenkolommen.

Sommige auteurs passen een computer toe voor stijlcorrectie, b.v. door te kijken naar de frequentie en plaats van bepaalde woorden ("types") in een tekst. Zelfs voor het construeren van een "plot" kan men een computer gebruiken, b.v. door te kijken of de butler in de eerste hoofdstukken niet opvallend vaak voorkomt.

De toepassing van computers voor tekstverwerking is zeer wijdverbreid.

Er zijn talrijke fabrikanten die computers leveren, waarop alleen programma's voor tekstverwerking beschikbaar zijn. De ontwikkelingen zijn nog in volle gang. Het is b.v. de vraag of het handig is van een tekst die continu opvraagbaar is op een beeldscherm ook nog een versie op papier te vervaardigen. Wel is het vaak weer handig b.v. op kantoren om bij een notitie een commentaar in te kunnen spreken. Dat commentaar kan gedigitaliseerd worden en opgeslagen, om later weer te worden afgespeeld. Het klassieke "qwertyuiop" toetsenbord is soms ook in discussie. In de toekomst zal het waarschijnlijk mogelijk worden teksten rechtstreeks aan de computer te dicteren.

### 8.5.3.3 Leermachines

Leermachines ("teaching machines") waren al in de handel in de zestiger jaren zonder dat van digitale computers gebruik werd gemaakt. Het was een gemechaniseerde vorm van de reeds lang bekende "geprogrammeerde instructie". Bij geprogrammeerde instructie zijn de uitspraken die een leerling aan een leerling wil voorleggen van te voren opgesteld. Een uitspraak of een reeks uitspraken wordt steeds afgesloten met een vraag aan de leerling, dikwijls in meerkeuzevorm. Als de leerling de vraag foutief beantwoordt wordt hij naar een andere uitspraak verwezen, dan wanneer hij de vraag juist beantwoordt. In het eerste geval wordt b.v. uitgelegd waarom het door de leerling gegeven antwoord onjuist is. Leerlingen die de vragen steeds juist beantwoorden gaan sneller door de stof dan leerlingen die vragen foutief beantwoorden. Als voordelen van geprogrammeerde instructie worden wel genoemd, dat de leerlingen aldus hun eigen leersnelheid kunnen bepalen en de directe terugkoppeling ("feed back") die zij op hun antwoorden krijgen. Als nadeel kan gelden, dat het ontwerpen van zo'n geprogrammeerde instructie een arbeidsintensief werk is.

De geprogrammeerde instructie kan men automatiseren, zonder gebruik te maken van computers, b.v. door de uitspraken op microfiche te plaatsen, tesamen met eenvoudige stuurinformatie die aangeeft naar welk volgende beeld moet worden overgeschakeld bij de verschillende alternatieven die de leerling kiest. De projector voor de microfiches bevat dan een simpel sturingsmechanisme dat een nieuw beeldje kiest, afhankelijk van de door de leerling ingedrukte knop (b.v. een met "ja" en een met "nee") en de op de microfiche opgenomen stuurdata.

Zulke leermachines kan men met behulp van digitale computers eenvoudig namaken, waarbij de mogelijkheden gemakkelijk vergroot kunnen worden, b.v. met betrekking tot het aantal mogelijke antwoorden en de complexiteit van het pad dat de leerling door de leerstof kan volgen.

Dit soort onderwijs per computer wordt vaak aangeduid met de afkorting COO (voor "computerondersteund onderwijs"), of met CAI (voor het engelse "computer aided instruction"). De computer kan hierbij ook gemakkelijk gebruikt worden voor het vervaardigen van de lessen en daarvoor zijn dan ook speciale programma's geschreven.

Wanneer een leraar zo'n programma gebruikt om de lessen te maken, moet hij de uitspraken die hij aan de leerlingen wil voorleggen in de computer brengen, en daarbij van tevoren een structuur kiezen, de volgorde waarin de uitspraken aan de leerlingen zullen worden gepresenteerd. In sommige gevallen kan de leraar het aan de computer overlaten welke volgorde wordt gekozen. Een enigszins triviaal voorbeeld is een programma om woorden en uitdrukkingen in een vreemde taal te leren. In een bestand is een lijst van woorden met de vertaling ingebracht en de computer legt deze een voor een in een willekeurige volgorde aan de leerling voor. Men kan de computer ook programmeren om de "vorderingen" van de leerling bij te houden en/of de leerling verbaal te stimuleren door opmerkingen als "zeer goed gedaan!" op het scherm te laten verschijnen.

Het systeem gaat nog meer op een leraar lijken, als de leerling ook vragen kan stellen aan de computer. In het voorbeeld van de woorden in een vreemde taal zou dat te realiseren zijn door van hetzelfde bestand gebruik te maken, wanneer het bestand aan woorden met hun vertaling maar groot genoeg is om aan de vraagdrift van de leerlingen te beantwoorden. In rekenen en logica zijn computers "van huis uit" tamelijk goed en het is dan ook betrekkelijk eenvoudig programma's te vervaardigen die een leerling rekenen kunnen leren, zonder dat de leraar van tevoren alles heeft moeten bedenken. Men zou zo'n programma een "expert system" in rekenen kunnen noemen: vrijwel alle rekenproblemen die iemand kan bedenken kunnen sommige computers oplossen. (Over "expert systems" in het algemeen: zie hieronder.)

#### 8.5.4 Accent Op Het Verplaatsen Van Data

##### 8.5.4.1 Teletekst

Teletekst is een voorbeeld van het gebruik van zend- en ontvanginginstallaties voor elektro-magnetische signalen voor het overbrengen van digitale informatie. Bij teletekst wordt digitale informatie "meegezonden" met het normale televisiesignaal. Een speciale elektronische "decoder", ingebouwd in of verbonden aan een t.v.-toestel, zorgt ervoor dat steeds een pagina tekst op het scherm wordt weergegeven. Gebruikers kunnen de gewenste pagina kiezen d.m.v. een toetsenbord.

Hoewel op het eerste gezicht de communicatie wat eenzijdig lijkt omdat de ontvangers geen informatie naar de zenders kunnen teruggeven zijn de mogelijkheden van teletekst zeer groot. Het systeem kan dienst doen als elektronische krant, waarvan er iedere dag zeer veel verschillende edities kunnen worden verzorgd en verspreid. De tekst kan ook over het gewone t.v.-signaal heen worden weergegeven en dan kan teletekst b.v. ondertitels verzorgen: de krant brengt iedere keer als een nieuwe ondertitel moet verschijnen een nieuwe editie uit (die alleen voor de betreffende pagina verschillend hoeft te zijn).

Men kan aan het decodeerapparaat ook een extra computer verbinden die de ontvangen data, al dan niet na selectie, opslaat. De ontvangen data kunnen ook bestaan uit computerprogramma's, zoals een experiment van de B.B.C. in Groot-Brittanie laat zien (in Nederland via de radio m.b.v. de N.O.S.-"basicode"). Men zou zo'n computer ook een videorecorder kunnen laten sturen, zodat men voor automatische opname van een bepaald t.v.-programma niet meer afhankelijk is van het precieze begintijdstip.

Teletekst zou niet alleen gebruikt behoeven te worden voor het verzenden van boodschappen die voor iedereen bedoeld zijn. Voor berichten die maar voor een beperkt aantal personen bedoeld zijn kan gebruik worden gemaakt van moderne geheimschrifttechnieken met de z.g. valluikcodering (vgl. Verhoeff, 1980, blz. 251). Deze technieken geven ook een oplossing voor het probleem van de authenticiteit van een bericht, wat van belang kan worden wanneer gebruikers ook advertenties en/of ingezonden brieven kunnen plaatsen in teletekst.

#### 8.5.4.2 Viewdata

Dit is een voorbeeld van een systeem, waarbij de gewone telefoonverbindingen worden gebruikt voor communicatie met computers. Het is in het algemeen mogelijk digitale data te moduleren naar een vorm die zich via telefoonlijnen laat verzenden. Aan de ene kant bevindt zich een modem (een apparaat voor modulatie, van analoog naar digitaal, en demodulatie, van digitaal naar analoog), waaraan een computer is aangesloten, en aan de andere kant een modem met een beeldscherm en toetsenbord.

Bij Viewdata (in Nederland onder de naam "Viditel", in Groot-Brittannie onder de naam "Prestel" geëxploiteerd) zijn de mogelijkheden qua apparatuur en programmatuur gestandaardiseerd. Zoals altijd levert dat een aantal voordelen, maar ook z'n beperkingen op. De toepassingen worden op dit moment voornamelijk gezocht in het verschaffen van teksten en eenvoudige tekeningen, al dan niet van en voor beperkte groepen. Voor het vinden van de gewenste "bladzijden" (de inhoud van een scherm) wordt gebruik gemaakt van een "boomstructuur", waarbij de gebruikers de gewenste bladzijde moeten vinden door aan de hand van korte omschrijvingen steeds de juiste "tak", "sub-tak" etc. te vinden. (Zie verder voor de mogelijkheden en beperkingen van teletekst en viewdata b.v. Viewdata en Videotext, 1980.)

#### 8.5.4.3 Datanetten

Behalve via het gewone telefoonnet kunnen computers met randapparatuur ook met elkaar worden verbonden via speciale kabels. De betrouwbaarheid en de snelheid van de verbinding kunnen daardoor vergroot worden. Wanneer de verbindingen een zekere complexiteit hebben, d.w.z. meer dan twee eindpunten hebben, kan men van een "net" spreken. Zo'n

net kan zich b.v. in een gebouw bevinden, en men noemt het dan wel - enigszins arbitrair - "lokaal". Allerlei fabrikanten brengen hun eigen faciliteiten voor datanetten in de handel, met hun eigen "standaarden" voor apparatuur en programmatuur daarvoor. Zulke netten hebben, al dan niet in aansluiting op de mogelijkheden om per telefoon digitale data uit te wisselen, "gedistribueerde dataverwerking" mogelijk gemaakt. Hierbij zijn verschillende computers op elkaar aangesloten, en kunnen ze over elkaars data beschikken.

Sinds 1983 is er in Nederland ook een niet lokaal datanet in gebruik genomen. Het wordt onderhouden en beschikbaar gesteld door de P.T.T.. In beginsel verschaft dit net snellere, goedkopere en meer betrouwbare verbindingen dan het telefoonnet aan computergebruikers kan bieden.

Algemeen kan men zeggen, dat datanetten een ontwikkeling van een "eerste generatie"- techniek van aan elkaar koppelen van computers naar een "tweede generatie"- techniek vormen. Bij de eerste generatie computers werden immers de componenten (hier de computers) achteraf met elkaar verbonden, terwijl bij de tweede generatie de onderlinge verbindingen eerst werden gelegd, terwijl de componenten later werden toegevoegd.

#### 8.5.5 Combinaties

##### 8.5.5.1 Bestuurlijke informatiesystemen

De term waarmee de hier bedoelde systemen worden aangeduid is evenals de Engelse vertaling, "management information systems", enigszins misleidend. Met "bestuurlijk" wordt in dit verband gewoonlijk niet bedoeld dat zulke informatiesystemen niet gebruikt zouden worden door diegenen in een organisatie die "operationeel" zijn, die het werk uitvoeren, in tegenstelling tot hen die de organisatie besturen. Op ieder "niveau" in organisaties worden beslissingen genomen en bestuurlijke informatiesystemen hebben de pretentie informatie te verschaffen die nodig is voor het nemen van die beslissingen.

Aan de andere kant is het bij bestuurlijke informatiesystemen wel zo, dat in ieder geval ook de leiding van een organisatie er gebruik van maakt en dat informatie die belangrijk is voor het besturen van een organisatie ook in het systeem is opgenomen. Door een systeem een bestuurlijk informatiesysteem te noemen geeft men dan ook aan dat het systeem in een organisatie "geïntegreerd" is. Bij ingewikkelde organisaties zullen de bestuurlijke informatiesystemen dus gewoonlijk tamelijk complex zijn, dus b.v. vele in- en uitvoer- en verwerkingsmogelijkheden hebben met diverse onderlinge communicatiemogelijkheden, al dan niet via een centrale computer.

Geautomatiseerde bestuurlijke informatiesystemen vormen een combinatie van de meeste hierboven beschreven mogelijkheden, op een onder-



ling samenhangende manier. In de rechtspraak, zowel bij wetgeving, rechtspraak en uitvoerende macht als bij niet-publiekrechtelijke organisaties als rechtshulp, reclassering e.d. zullen geautomatiseerde bestuurlijke informatiesystemen waarschijnlijk een belangrijke rol gaan spelen. In de organisatie van gemeentebesturen in Nederland wordt op veel plaatsen in een hoog tempo geautomatiseerd, hetgeen in sommige gevallen reeds geleid heeft tot tamelijk omvangrijke geautomatiseerde bestuurlijke informatiesystemen. Tot volledig geïntegreerde systemen, waarbij alle gemeentelijke dataverwerkende taken, van het bijhouden van personenregisters, het verwerken en bewaren van gemeentelijke verordeningen tot het behandelen van in- en uitgaande en interne stukken van het gemeentelijke apparaat, per computer geschiedt is het naar mijn weten nog nergens gekomen.

Met betrekking tot de wetgevende taken op ministeries is er ook een mogelijke toekomst voor bestuurlijke informatiesystemen. Op den duur is denkbaar, dat ministers na kunnen gaan hoever ieder wetsontwerp of ontwerp van een koninklijk besluit is gevorderd, en welke ambtenaren daarmee bezig zijn of bezig geweest zijn (al zullen daarvoor waarschijnlijk enige organisatorische wijzigingen ten departemente dienen plaats te vinden).

Voor openbare ministeries en gerechtshoven zijn er buiten Nederland, met name in de V.S., diverse bestuurlijke informatiesystemen ontwikkeld. Sinds 1971 is in Washinton D.C. t.b.v. de vervolging in strafzaken een systeem in werking, "Promis" ("prosecution management information system") genoemd, met de volgende doelen (vgl. Promis, 1975, blz. II.2):

1. Het mogelijk maken bij de voorbereiding van de zaken middelen aan te wenden overeenkomstig het belang van de zaak.
2. Het mogelijk maken van een eerlijke en consistente toepassing van het opportuniteitsbeginsel.
3. Het mogelijk maken de planning voor het afhandelen van zaken te beheersen en te vergemakkelijken.
4. Het mogelijk maken problemen bij het behandelen en vervolgen van strafzaken te analyseren en te onderzoeken.

Om deze doelen te verwezenlijken is o.a. een uitgebreid netwerk van computerterminals verspreid over het district Washington opgebouwd.

Tenslotte geef ik een wat uitvoeriger overzicht van een verzameling bestuurlijke informatiesystemen in de V.S., ontwikkeld voor de min of meer los van het "gewone" justitiele systeem opererende jeugdrechtshoven (vgl. Boxerman, 1977). Het betreft een omvangrijk projekt van de Amerikaanse "National Council of Juvenile and Family Court Judges", in 1977 voltooid.

Het project werd "JISRA" genoemd: "Juvenile Information System Requirements Analysis". Het bestond uit twee fasen:

1. het inventariseren en beoordelen van de bestaande systemen alsmede de bestaande behoeften,
2. het ontwikkelen van een standaardsysteem.

Er worden vijf behoeften en vereisten onderscheiden waaraan het systeem moet voldoen:

1. vereisten, voortvloeiend uit de gerechtelijke en administratieve procedures
2. behoeften van degenen die dagelijks met het systeem werken
3. behoeften van statistische aard
4. behoeften van de leiding inzake planning, beheer en begroting
5. behoeften van wetenschappelijke aard en van evaluatie.

Het "model-systeem" dat het resultaat is van het project heeft de volgende faciliteiten:

1. het bijhouden van de documentatieregisters (interne administratie),
2. het behandelen van de afzonderlijke zaken,
3. het besturen van de organisatie,
4. het verschaffen van rapporten voor de buitenwereld en
5. het voorzien in enkele speciale behoeften m.b.t. onderzoek en evaluatie.

#### 8.5.5.2 Computeradviessystemen en verdere ontwikkelingen

Ontwikkelingen in het veld van de "kunstmatige intelligentie" ("artificial intelligence"), waarop ik hieronder nader inga, hebben geleid tot verdere uitbouw van de mogelijkheden van computers, waarvan leermachines en bestuurlijke informatiesystemen reeds voorbeelden vormden. Met kunstmatige intelligentietechnieken is het volgens de beoefenaren onder bepaalde voorwaarden mogelijk kennis te representeren in de vorm van data. Computerprogramma's die bekend zijn geworden onder de naam "expertsytemen" ("expert systems") verschaffen de mogelijkheid om uit deze data nieuwe data af te leiden, die door mensen ook weer als kennis geïnterpreteerd kunnen worden. Een hulpmiddel hierbij is een program-

maonderdeel ("natural language interface" genaamd) dat de data doet verschijnen in de vorm van eenvoudige volzinnen in natuurlijke taal.

Een bekend voorbeeld is het "MYCIN"-systeem (vgl. Boden, 1981, blz. 440), een interactief programma dat een medisch adviseur, gespecialiseerd in besmettelijke ziekten, simuleert. Het programma verzorgt een vraag en antwoordsituatie (die gemiddeld zo'n 20 minuten duurt) met artsen die de hulp van een specialist nodig hebben. In 75% van de gevallen zou het programma tot hetzelfde oordeel komen als een menselijke deskundige. De arts vraagt het Mycinsysteem om advies bij de identificatie van micro-organismen en de voor te schrijven antibiotica alsmede om uitleg van dat advies op het gewenste niveau van gedetailleerdheid.

Een ander, vergelijkbaar voorbeeld is "Explorer", dat in staat zou zijn uit data, bevattende kennis over geologische aspecten van een bepaalde streek, te voorspellen welke delfstoffen in de streek met welke waarschijnlijkheid aangetroffen zouden kunnen worden.

Het "Mycin"-programma is in de handel verkrijgbaar zonder de data, de kennis die het een medisch expert maken, nl. als "Emycin" ("empty Mycin") dat "gevuld" kan worden met andere kennis, eveneens d.m.v. interactieve processen met eenvoudige (Engelse) natuurlijke taal. Ook bestaan er programma's, geschikt voor sommige microcomputers, die ongeveer op dezelfde manier werken als "Emycin".

Expertsystemen werken niet louter met "harde" statistische waarschijnlijkheden. Ook het intuïtieve oordeel van degene die het kennisbestand opbouwt wordt in de konklusies betrokken. De systemen kunnen dan ook in veel gevallen verschillende (beargumenteerde) oordelen geven naast het oordeel dat als het meest waarschijnlijke uit de berekeningen komt.

Zie ik het wel, dan zou men de gehanteerde algoritmen "convergent" kunnen noemen in de zin, dat uit een grote hoeveelheid data een beperkt aantal verschillende konklusies kunnen worden afgeleid, zij het ieder met een verschillende waarschijnlijkheid en/of intuïtieve aannemelijkheid. Het "divergent" combineren van bepaalde mogelijkheden tot een geheel nieuw pakket b.v. van te nemen maatregelen lijkt met deze systemen (nog) niet mogelijk.

T.a.v. de mogelijkheid om expertsystemen te gebruiken in het recht merkt Boden (1981, blz. 441) op dat deze in dat geval "subtielere begrippen" dan (Bayesiaanse) waarschijnlijkheid, zoals "vertrouwen" ("confidence") en "bewijs" ("evidence") moeten bevatten om rechterlijke absurditeiten van verschillende aard te vermijden. Ook deelt zij t.a.p. mee de eerste prototypen voor juridische toepassing van dit soort systemen pas in 1988 te verwachten.

Computeradviessystemen kunnen al dan niet gebruik maken van de hier

beschreven ("convergente") technieken van de nu gangbare expertsystemen. In de toekomst zullen dit soort programma's waarschijnlijk een combinatie van verschillende mogelijkheden van representatie van kennis in de vorm van data bevatten. Een mogelijke aanzet is de "Computer-Based Consultant" (Hart, 1975), een systeem dat o.a. leerling-monteurs adviezen kan geven over het in elkaar zetten van bepaalde machines. Een typisch voorbeeld van "divergente" techniek geeft Sprowl (1980, blz. 195). Het "Legol"-systeem, dat door Stamper ontwikkeld wordt aan de London School of Economics (Stamper, 1980, blz. 45) is een combinatie van "klassieke" database- en van kunstmatige intelligentie technieken. Het project moet leiden tot een uitgebreid computeradviesstelsel op juridisch terrein.

In hoofdstuk 12 wordt een project beschreven, dat een computeradviesstelsel tot resultaat had op het terrein van de straftoemeting.

Naar mijn mening ligt de zwakte van al deze systemen op dit moment in het samenstellen van het databestand dat kennis representeert. Bij de opbouw daarvan is als gevolg van de semantische vraagstukken die bij kennisrepresentatie een rol spelen, menselijke tussenkomst onontbeerlijk. In de toekomst zullen waarschijnlijk meer en meer pogingen worden ondernomen om programma's te maken die uit "gewone" databestanden, b.v. die van bestuurlijke informatiesystemen, databestanden afleiden, die meer algemene kennis inhouden m.b.t. de feiten waarover die data informatie bevatten. (Men spreekt in dit verband wel van "machine learning".) Het is de vraag in hoeverre en hoe snel die pogingen resultaat zullen opleveren, omdat de semantische problemen waarschijnlijk niet geringer zullen zijn dan die bij het automatisch vertalen.

## 8.6 KUNNEN COMPUTERS DENKEN?

Een van de pioniers in de informatica, Alan Turing, stelde de volgende "test" voor ter beantwoording van de vraag "Kunnen machines denken?" (Vgl. Hofstadter, 1980, blz. 595). Wanneer iemand, die per telex communiceert met iemand anders na verloop van enige tijd niet door heeft dat aan de andere kant geen persoon zit, maar dat een machine de conversatie voert, dan kan die machine denken. Reeds in de beginjaren van de digitale computertechniek voldeed het computerprogramma "Eliza" van Weizenbaum aan deze test, zij het, dat dit in belangrijke mate samenhang met de naïviteit van de testpersonen. Niettemin stelden sommigen en onder hen Weizenbaum voor, de test te verzwaren. Voorstellen die wel gedaan zijn, bestaan uit het eisen, dat de persoon in de test een computerdeskundige of een vooraanstaand wetenschapper is. Ook wordt (vgl. Weizenbaum, 1980, blz. 566) wel de eis gesteld dat een computer eerst maar eens met "originale ideeën" moet komen.

Korte tijd nadat Euwe in zijn oratie te Rotterdam de vraag stelde of computers konden denken, stelde Frijda (1965) de dan ook relevante vraag "Kunnen mensen denken?". Het antwoord op zulke vragen hangt natuurlijk af van de vraag, naar welk begrip de term "denken" verwijst.

Gelijksoortige discussies kan men voeren over de de term "kunstmatige intelligentie" of "machine-intelligentie". Onder psychologen hebben validiteit en betrouwbaarheid van intelligentietests jarenlang voorwerp van heftige polemieken uitgemaakt en voor sommigen wordt de zaak nog opwindender wanneer machines mede-belanghebbende zijn.

Wat tot de tak van wetenschap die gewoonlijk wordt aangeduid als "kunstmatige intelligentie" wordt gerekend is vnl. historisch bepaald. De beoefenaren van deze tak hielden zich bezig met bepaalde vraagstukken o.a. het vertalen per computer en het spelen van schaak door een computer. Voor het oplossen van deze en dergelijke vraagstukken werden allerlei technieken, waaronder ook speciale computertalen vervaardigd. Die technieken beogen veelal computers werkzaamheden te laten verrichten die, zo ze door mensen zouden worden verricht, "cognitieve vaardigheden" zouden vereisen. Soms is de benadering eerst te kijken hoe mensen iets doen dat cognitieve vaardigheden vereist en dan de computer te programmeren zodat hij het zo goed mogelijk nadoet. Een andere benadering is alleen naar het resultaat van het menselijke gedrag te kijken en de computer alleen op het resultaat te programmeren.

Een interessantere vraag dan de vraag of computers kunnen denken, en waarmee ik dit hoofdstuk wil besluiten is de vraag waarom computers sommige cognitieve vaardigheden maar zo slecht beheersen en andere juist heel goed. Vertalen in natuurlijke talen kunnen computers op dit moment heel slecht (hoewel daar veel mankracht in is gestoken), schaken kunnen ze zo langzamerhand vrij goed, al zijn ze nog niet van topklasse (vgl. Van den Herik, 1983). Computers kunnen wedijveren met medische specialisten in de diagnose, en zijn vaak betere leraren dan de menselijke, maar een automatische stofzuiger met voldoende intelligentie om geen briljantje, maar wel broodkrumels op te zuigen moet nog gemaakt worden.

Het antwoord op deze vraag hangt naar mijn mening af van een goede theorie over cognitieve vaardigheden. Waarschijnlijk kan het model van Guilford over het menselijk denken (vgl. Meuwese, 1973, blz. 199) de vraag voor een goed deel beantwoorden. De naar complexiteit geordende operaties (cognitie, geheugen, divergente produktie, convergente produktie en evaluatie) geven een aanwijzing hoe moeilijk computers voor een bepaalde taak geprogrammeerd kunnen worden. Een en ander is natuurlijk afhankelijk van het aantal ingezette manuren van programmeurs van voldoende niveau en van de produkten en inhouden uit het model van Guilford. Mijn konklusie hieruit is tenslotte niet, dat leraren en artsen niet intelligent zijn, maar wel dat ze voor het uitoefenen van hun dagelijkse

taak hun cognitieve vaardigheden nauwelijks gebruiken. In de praktijk blijkt ook, dat b.v. artsen die niet de volgens sommigen vereiste diploma's bezitten in ziekenhuizen jarenlang succesvol "hun" vak kunnen uitoefenen. De "strategische beleidsvorming", waartoe computers volgens Verhoeff nog niet in staat zijn, is waarschijnlijk van het niveau "evaluatie" uit het model van Guilford.