26
A 59

De Computer in de Letteren

UABF
003
654

outerafdeling Faculteit der Letteren / Universiteit van Amsterdam

BIBLIOTH. AMSTELAED.

lc 8110-1303 1

125

XXVI A 59

"De computer in de letteren".

Inhoud.

0	Voorwoord.	2
1	Iets over het ontstaan van de computer-afdeling.	3
2	Mogelijkheden en beperkingen van een computer. Waarom een computer-afdeling nodig is.	5
3	Waar kan onze faculteit een computer voor gebruiken ?	7
4	Wat is er zo al gedaan in onze faculteit ?	11
5	Welke ervaring en standaard-programmatuur hebben wij nu ?	34
6	Onderwijs op de computer-afdeling.	42
7	Hoe kom ik aan de slag ?	44
8	Bibliotheek en tekst-bestanden.	45
9	Apparatuur en programmeertalen.	46
10	Index.	47

Taalwetenschap
Literatuurwetenschap

Uitgegeven door:

Computer-afdeling van de Faculteit der Letteren,
Universiteit van Amsterdam,
Herengracht 338, (onder het Fonetisch Instituut),
1016 CG Amsterdam.
Telefoon: 020 - 5252181.

Amsterdam, april 1981.

Staf:

T. Muller
drs.J. Skolnik
ir.G.J. van der Steen

Druk: drukkerij Kaal bv, Amsterdam.

Instituut voor Theaterwetenschap
Nwe. Doelenstraat 16
Amsterdam-C.

De illustratie op de omslag is een gedetailleerde vorm van het Hettische teken voor "leven", zoals dat gebruikt wordt om computer-output als op pagina 19 te krijgen.

Hoofdstuk 0. Voorwoord.

In dit boekje laten wij u kennismaken met de computer-afdeling van de Faculteit der Letteren van de Universiteit van Amsterdam. Het is geschreven voor twee verschillende groepen mensen, wat typerend is voor het werk dat in de computer-afdeling wordt gedaan. De eerste groep is de wetenschappelijke staf van onze faculteit. Velen daarvan zullen zich al een oordeel hebben gevormd over de wenselijkheid van computer-gebruik in hun vakgebied. Wij willen in dit boekje echter laten zien dat de stap tot het daadwerkelijk inschakelen van de computer waarschijnlijk veel kleiner is dan u denkt. De direkte aanleiding daartoe is het bestaan van onze computer-afdeling, die sinds mei 1977 functioneert. De aktiviteiten daarvan zijn zo omvangrijk geweest, dat dit boekje veel later is verschenen dan in de bedoeling lag. Aan de andere kant biedt dit het voordeel dat rapportage mogelijk is over een aantal projekten die grotendeels met behulp van de computer-afdeling zijn gerealiseerd.
Veel toepassingen hebben algemene trekken. We hebben daarom een overzicht opgenomen van computer-toepassingen zoals die regelmatig worden gepubliceerd. Hierdoor kunt u een kijkje nemen hoe het bij anderen toegaat. Dit geldt in het bijzonder voor de tweede groep waarvoor wij dit boekje uitbrengen: de belangstellenden buiten onze faculteit die vaak nieuwsgierig zijn waarom wij een computer gebruiken. Voor onze vakbroeders in de informatica is af en toe een extra regeltje geschreven. We hopen hen te laten zien dat er in onze faculteit vele interessante computer-toepassingen zijn die verder onderzoek waard zijn.
Het is de bedoeling dat op onregelmatige tijden een vervolg van dit boekje zal uitkomen, waarin nieuwe programma's en gebruiks-mogelijkheden worden beschreven en verslag gedaan zal worden van lopende- of afgesloten projekten.
Wij danken iedereen die een bijdrage aan dit boekje leverde. De teksten die niet zijn ondertekend zijn door de samensteller geschreven, waarbij J. Skolnik bijdragen leverde aan de hoofdstukken 4, 5 en 6. Van hem is ook het ontwerp van de letter. De opmaak werd verzorgd door T. Muller, daarbij geassisteerd door het tekst-opmaak programma "Scribe."

Voor de computer-afdeling,
G.J. van der Steen, samensteller

Amsterdam,
februari 1981

Ic 8110-1303

Hoofdstuk 1. Iets over het ontstaan van de computer-afdeling.

We gaan terug naar het najaar van 1974. Verschillende onderzoekers uit de letterenfaculteit waren al met computer-verwerking in aanraking gekomen; in de meeste gevallen was gebruik gemaakt van de diensten van het Mathematisch Centrum. Er stond ook duidelijk wat op stapel: voor het projekt Hettitisch was een software-huis in de arm genomen. Zelfs op het gebied van het onderwijs was er toen al iets gebeurd: Brandt Corstius had studenten Neerlandistiek tot het enthousiast schrijven van programma's gebracht. Maar toch was alles nog incidenteel en/of individueel: van iets gerichts in de faculteit was geen sprake. De tijd leek rijp om daar iets aan te gaan doen, hoewel nog niet duidelijk was hoe. Aan computer-apparatuur was in de FdL nagenoeg niets aanwezig. Er was bijvoorbeeld geen enkele aansluiting op de computer van SARA. Het enige was de inmiddels bijna 10 jaar oude, kleine en beperkte IBM 1130 van de vakgroep Fonetische Wetenschappen. Aan vervanging daarvan was eigenlijk nog nooit serieus gedacht, hoewel de machine zo langzamerhand toch wel verouderd was. Door een min of meer toevallige samenloop van omstandigheden werd duidelijk, dat het mogelijk zou moeten zijn de oude computer te verkopen, en te vervangen door "iets nieuws". Binnen Fonetiek is toen het idee ontstaan om van dat "iets nieuws" iets te maken, dat niet alleen door de eigen vakgroep, maar ook door de rest van de faculteit gebruikt zou kunnen worden. Hoewel een computer als vast onderdeel van de apparatuur voor een fonetisch instituut beschouwd mocht worden, was ook bekend dat een dergelijke computer een zeker percentage leegloop heeft. Met een geschikte keuze van de apparatuur, in een qua randapparatuur iets uitgebreidere configuratie, waarmee bovendien de koppeling met de computer van SARA (het gezamenlijk rekencentrum van de Amsterdamse universiteiten en het Mathematisch Centrum) onderhouden zou kunnen worden, moest het mogelijk zijn de gehele faculteit van de apparatuur te laten profiteren. Na enig overleg werd door de Faculteitsraad in februari 1975 de Commissie Computer-faciliteiten (COCOFAC) ingesteld, die tot taak kreeg de behoeften in de faculteit nader te onderzoeken, en eventueel met voorstellen voor de aanschaf van apparatuur te komen. Hoewel door deze gang van zaken de ontwikkeling nogal "vanuit de apparatuur" plaatsvond, gebeurde er meer waardoor de zaak bijna vanzelf groeide in de richting van een gezamenlijke aanpak voor de gehele faculteit.
Begin 1975 werd door het College van Bestuur de Centrale Informatica Commissie (CIC) ingesteld, bestaande uit vertegenwoordigers uit elke (inter-)faculteit. De letterenfaculteit wees Kwee Tjoe Liong als haar vertegenwoordiger aan.
De werkzaamheden van de CIC hebben onder andere geresulteerd in de oprichting van de Interfacultaire Werkgroep Informatica (IWI). In zo'n werkgroep kunnen alleen vakgroepen deelnemen; dat zijn vanuit onze faculteit Algemene Taalwetenschap en Fonetiek, want de computer-afdeling heeft, zoals verderop zal blijken, geen officiële status, zeker niet die van "vakgroep". Maar de afspraken zijn duidelijk: de deelnemers in IWI stellen zich op als facultaire vertegenwoordigers.
In oktober 1975 verscheen een eindverslag van de COCOFAC met onder andere als aanbevelingen:
- om voor de coördinatie in alle vraagstukken met betrekking tot informatica en computer-gebruik een (vaste) Computer-commissie in te stellen,
- om Emiel Kappner alvast als systeembeheerder aan te stellen.

De faculteitsraad nam de voorstellen over, en de bestelling van de apparatuur werd doorgestuurd naar het CvB. Dat laatste klinkt eenvoudig, maar in de tussentijd was een enorme hoeveelheid werk verzet: marktverkenning, vergelijking en keuze van apparatuur met alles wat daaraan vastzit (onder andere systeem-programmatuur), behandeling van de aanvraag door ASTUR (adviescommissie van het CvB) en CRIVA (adviescommissie van de minister); de uiteindelijke goedkeuring door de staatsecretaris was in oktober 1975 al binnen. Het zal duidelijk zijn dat dit karwei in zo'n korte tijd alleen maar gedaan kon worden met de hulp en inzet van velen. Zonder iemand te kort te willen doen, wil ik er een paar met name noemen:
- L. Boves (Instituut voor Fonetiek, Nijmegen), van wiens marktonderzoek dankbaar gebruik gemaakt kon worden, die geen moeite te veel was, en zonder wiens hulp de technische kant nooit zo snel rond was gekomen.
- H. van Eerde (coördinator automatiserings-zaken), die vanuit zijn positie de hele gang van zaken voortreffelijk heeft begeleid.

Na de snelle start leek alles opeens wat moeizamer te gaan, maar in juni 1976 was het dan zover: op Herengracht 338 werd een computer naar binnen getakeld. Weliswaar was de apparatuur nog niet compleet, maar het basis-systeem was er, en er kon gestart worden. Eigenlijk: het begin van de "computer-afdeling" was er. Toen kwam er weer zo'n gelukkige samenloop van omstandigheden, we werden in de gelegenheid gesteld om extra personeel aan te trekken. De hele sollicitatie-procedure kostte (oormerking, interne werving, enzovoorts) nogal wat tijd, maar in mei 1977 kwam Gert van der Steen, een maand later gevolgd door Joos Skolnik. Op dat moment was ook daadwerkelijk de mankracht aanwezig om schot te krijgen in de verschillende onderzoeks-projekten. In mei 1978 kwam ook Tineke Muller nog (op een vrijgekomen plaats van Fonetiek), die de zorg draagt voor het dagelijkse reilen en zeilen rond de computer, een niet te onderschatten factor binnen zo'n bedrijf.
Al met al was de zaak daarmee uitgegroeid tot een rekencentrumpje. In het voorjaar 1977 zijn we de term "computer-afdeling Faculteit der Letteren" gaan gebruiken om duidelijk te maken dat dit een instelling was voor dienstverlening en ondersteuning van de gehele facultaire gemeenschap. Een officiële status (in de zin van de WUB) is er niet: het is geen vakgroep, geen werkgroep, geen instituut. Het enige officiële is: een eigen kostenplaats. De rest berust op afspraken; zo verrichten de medewerkers van de computer-afdeling hun

werkzaamheden onder toezicht en verantwoordelijkheid van de Computer-commissie, ook al behoren ze officieel tot de vakgroep Fonetische Wetenschappen (ze moeten toch ergens ondergebracht worden!). Misschien dat het gebrek aan officiële status soms nadelen heeft (het wat moeizaam van de grond krijgen van het informatica-onderwijs in de faculteit kan er mee te maken hebben), toch functioneert de computer-afdeling wel degelijk. Je zou kunnen zeggen: zolang de zaken in goed overleg geregeld kunnen worden, is er niets aan de hand. En zo gaat het dan ook.
De komst van de twee medewerkers bracht ook nieuwe kennis en ideeën in huis. Een gevolg daarvan was, dat het nodig bleek de hele verwerking van tekst-materiaal op het eigen mini-computer-systeem uit te voeren, en niet, zoals de aanvankelijke opzet was, alleen de in- en uitvoer en eventuele voor- en nabewerking.
Dat leidde tot de noodzaak van uitbreiding van vooral de schijf-capaciteit. De hele aanvraag-procedure had deze keer erg veel voeten in de aarde. Een tweede uitbreiding betreft de uitbouw naar een eigen, bescheiden, time-sharing systeem. Hiermee zal vanaf januari 1981 gewerkt kunnen worden.
De komst van de computer-afdeling ging voor de vakgroep Fonetische Wetenschappen niet ongemerkt voorbij. Een factor was: ruimte-problemen. Gelukkig stond het souterrain al enige tijd leeg, en daar zat een uitbreidings-mogelijkheid in. Inmiddels heeft de verbouwing plaatsgevonden, en is de apparatuur daarheen verhuisd. Misschien is niet iedereen met de nieuwe ruimte even blij (de hoogte is inderdaad wat beperkt), maar toch heeft de computer-afdeling er weer een stukje eigen status bij: een afzonderlijke ingang. We zullen zien of er nog zo'n mooi glimmend naambord afkan. Hoe dan ook: moge de computer-afdeling in haar half-ondergrondse toestand blijven wat het tot nog toe geweest is: een zoete inval, waar vogels van allerlei pluimage elkaar vinden om gezamenlijk tot een zinvol stuk werk te komen.

E.O. Kappner

Hoofdstuk 2. Mogelijkheden en beperkingen van een computer. Waarom een computer-afdeling nodig is.

Dankzij de snelle evolutie van computer-systemen zal deze brochure er over 10 jaar anders uitzien. Een aantal toepassingen zal echter hetzelfde blijven.
Het principe van een computer is dat je hem een programma kunt laten verwerken dat stapsgewijs aangeeft wat er met bepaalde gegevens moet gebeuren. Of dat nu heel snel gaat of langzaam, of er zeer veel gegevens verwerkt moeten worden of misschien zelfs geen, dit doet niets af aan het principe van het programmeren.
De mogelijkheden die het computer-geheugen biedt om in willekeurige vakjes iets op te bergen en later weer op te halen geeft een enorm potentieel voor de formulering van oplossingen voor een groot aantal problemen. Al meteen na de geboorte van de eerste computers stond men voor een overweldigend groot aantal mogelijkheden waaruit een jungle is ontstaan van programmeertalen en daarin geschreven programma's.
Bij het onderzoek naar de mogelijkheden van een computer werd al vanaf het begin gewerkt met een theoretisch model van een computer, de zogenaamde Turing-machine, waarop alle computers en alle programma's kunnen worden afgebeeld. Pas in de laatste jaren is er echter een theorie-vorming op gang gekomen over de vraag wat de inherente complexiteit is van een bepaald type probleem dat met de computer moet worden opgelost. Met andere woorden: hoe lang duurt het en hoeveel geheugen kost het minimaal of maximaal om dat type probleem op te lossen. Het wordt daarbij langzamerhand duidelijk dat, hoewel een computer erg veel kan, het in sommige gevallen toch heel lang zal duren voor hij de oplossing geeft. Dit neemt niet weg dat er een groot aantal problemen zijn die wel in een redelijke tijd zijn op te lossen, en in bepaalde gevallen is er een wedstrijd aan de gang om een veel gebruikt algorithme korter en sneller te maken. (Een algorithme is de formulering van de manier van oplossen van een probleem in een notatie die voldoende duidelijk is om er een programma bij te kunnen schrijven).
Kijken we naar de verwerking van een algorithme op een concrete computer dan is deze situatie nog rooskleuriger. Computers worden steeds sneller en goedkoper, zodat programma's die veel ruimte innemen of lang duren steeds minder problemen opleveren.
Om een programma op een computer te draaien zijn andere programma's nodig, de zogenaamde systeem-programmatuur. Deze zorgen ervoor, dat de programmeur en de gebruiker van diens programma's zo comfortabel mogelijk met de computer kunnen omgaan, dat wil zeggen zich zo min mogelijk behoeven te bekommeren om technische details die de constructie van de computer betreffen. De afgelopen 20 jaar hebben een grote evolutie te zien gegeven van deze systemen. Van de zogenaamde "batch-verwerking", waarbij mensen hun programma's aanbieden op ponskaarten en een dag later hun uitvoer mogen ophalen, tot aan geavanceerde "time-sharing" systemen toe, waarbij de gebruiker, achter een beeldscherm gezeten, met een grote computer kan converseren alsof hij deze geheel voor zichzelf heeft.
Het spreekt vanzelf, dat het voor die gebruiker heel wat plezieriger is om op die laatste manier de verrichtingen van zijn programma op de voet te volgen en direkt te kunnen reageren op uitvoer van de computer dan om met een batch-computer te werken. Om dit allemaal vlekkeloos te laten verlopen is echter een groot, gecompliceerd systeem nodig dat veel geheugen-ruimte vraagt.
Door de sterke prijsdaling is het echter tegenwoordig economisch verantwoord om voor gebruikers die niet te grote programma's interaktief moeten gebruiken eigen mini-computers te kopen en voor de nog kleinere programma's micro-computers. De prijzen van de laatsten zijn in de orde van enkele duizenden guldens.
Het grote voordeel is dat dan weer volstaan kan worden met een simpel bedrijfs-systeem, zoals zo'n 20 jaar geleden. Communicatie met andere gebruikers en het gebruik van bijzondere programmatuur zal men dan echter anders moeten regelen dan op een groot time-sharing systeem mogelijk is, bijvoorbeeld via een data-verbinding met de telefoon. Het programmeren en uittesten van een programma is een creatieve, maar ook een vrij technische aangelegenheid, waarbij in ieder geval een goede kennis van de gebruikte programmeertaal en het gebruikte bedrijfs-systeem nodig is. Heeft men deze niet, dan nog is het mogelijk om een algorithme te schrijven, namelijk in natuurlijke taal, waarbij men gebruik maakt van bepaalde gereserveerde woorden om de loop van het programma aan te geven. Op deze wijze hoort in ieder geval elk programmeren te beginnen.
Het zou wenselijk zijn dat iedereen die ooit eens een programma wil maken, of laten maken, zich bekwaamde in het formuleren van zijn probleem in een algorithmische vorm. Het algorithme is dan een simulatie van de manier waarop het probleem wordt opgelost. De tegenwoordige programmeertalen hebben zodanige constructie-mogelijkheden dat het algorithme dan eenvoudig in een programma is om te zetten. Communicatie met een programmeur is daardoor veel eenvoudiger omdat het probleem helder wordt geformuleerd door de enige die dat kan: de onderzoeker. In het onderwijs van de computer-afdeling wordt hieraan veel aandacht besteed.
Soms is het probleem eenvoudig maar ligt het accent op een andere vorm van communicatie: die met grote gegevens-bestanden (data-bases). Men is dan geïnteresseerd in het voorkomen van bepaalde relaties tussen die gegevens. Om het opvragen daarvan te vergemakkelijken worden zogenaamde vraagtalen ontwikkeld. Het streven is om deze vraagtalen zoveel mogelijk te laten lijken op natuurlijke taal. Het algorithme is daarbij geheel verdwenen: de vraagtaal wordt "non-procedural"; het systeem-programma dat de taal interpreteert stelt een aantal programma's in werking die zo efficiënt mogelijk het antwoord moeten leveren. Een dergelijke voor de gebruiker comfortabele situatie kan echter pas ontstaan na een jarenlang

speuren naar de kenmerken van diens essentiële behoeften: hoe zich zo kort en zo krachtig mogelijk te kunnen uitdrukken op zijn specifiek gebied.
En daarmee zijn wij op ons uitgangspunt terug: ondanks alle evolutie in computer-techniek, in bedrijfs-systemen, in programmeertalen, in data-base systemen zal de gebruiker die iets ervan gedaan wil hebben zijn probleem moeten formuleren. Voor de boekhouder is het nodig om precies op te schrijven wat hij doet als hij een post boekt. Dat valt hem niet zo moeilijk en daarom zijn, door de evolutie van alle computer-machinerie, er tegenwoordig zulke mooie administratieve systemen in gebruik.
Voor de filoloog is het nodig op te schrijven wat hij doet als hij de stam van een woord bepaalt. Voor de vertaler is het evenzo nodig om op te schrijven wat hij doet als hij vertaalt. Dit verklaart waarom het onderzoek naar machinaal vertalen in 1966 in de USA formeel werd gestopt: het bleek niet mogelijk om een vertaalproces te beschrijven zonder gebruik te maken van kennis omtrent de betekenis van woorden en zinnen. En hoe deze te beschrijven is nog steeds in een onrijp stadium van onderzoek. Bij zinnen hierover die beginnen met: "reeds nu is het mogelijk om....." dient men dan ook met een gezond wantrouwen verder te lezen.
In het algemeen zijn de computer-toepassingen in onze faculteit complexer en vereisen zij meer van programmeertalen en apparatuur dan toepassingen in de beta-faculteiten. Omgekeerd is door interesse en opleiding de letteren-onderzoeker vaak nauwelijks geïnteresseerd in de informatica. Dit leidt soms tot het verzamelen en in de computer invoeren van een groot aantal gegevens in het naieve vertrouwen dat daarna in enkele dagen de computer de meest fantastische analyses zal verrichten. Aan de andere kant zijn er door de computer betrekkelijk eenvoudige bewerkingen uit te voeren die voor de onderzoeker een grote taakverlichting kunnen betekenen. Wil men daarom het gebruik van de computer in een alfa-faculteit bevorderen dan zal voorlichting een belangrijke rol spelen.
Aan verschillende universiteiten, in binnen- en buitenland, is hieraan op verschillende manieren vorm gegeven. De meest onzakelijke vorm is die van de projekt-gewijze financiering waarbij van accumulatie geen sprake is (vooral in de USA gebruikelijk). Binnen de UvA is een inzicht gegroeid dat de communicatie tussen computer-staf en onderzoekers binnen de faculteiten moet gebeuren. Dit heeft, voor wat betreft de faculteit der letteren, geleid tot het oprichten van een computer-afdeling die eigen apparatuur beheert. In het licht van het voorgaande kunnen daarvoor de volgende taken worden aangegeven:

- ontlasten van de gebruiker van niet-relevante technische zaken
- voorlichting over (on)mogelijk computer-gebruik
- onderhoud en bijhouden van expertise op het gebied van hard- en software
- beheer van apparatuur
- ontwikkeling van veel gebruikte programma's
- cumuleren van ervaring: vergroting van inzicht in wat gemeenschappelijk nodig is en een overzicht van wat er gaande is
- contact met andere computer-afdelingen, uitwisselen van programmatuur en data
- eventuele uitbesteding van onderzoek of samenwerking met andere informatica vak- / werkgroepen.

Hoofdstuk 3. Waar kan onze faculteit een computer voor gebruiken ?

Als voorbeelden zullen we een aantal goedlopende- of reeds afgesloten projekten aanduiden, zoals die recentelijk in de literatuur of op congressen van de volgende organisaties is gerapporteerd:
a. Association for Computational Linguistics. Tijdschrift: AJCL.
b. Association for Literary and Linguistic Computing. Tijdschrift: ALLC-bulletin.
c. Tijdschrift: Computers in the Humanities.
d. 2-Jaarlijks congres: Coling (Computers and Linguistics).

Een vrij algemene indeling die daarbij wordt aangehouden is de volgende:

- In- en Uitvoer
- Lexicografie / Lexicologie
- Grammatica's
- Semantiek
- Taalkunde
- Taalverwerving
- Documentatie
- Archeologie
- Geschiedenis
- Musicologie
- Fonetiek.

In- en Uitvoer.

Vooral voor filologische toepassingen is de in- en uitvoer van de computer erg belangrijk.
a. Invoer.
Om teksten te kunnen behandelen moeten ze op de een of andere manier in het geheugen van de computer worden gebracht. Een filoloog wil het liefst dat alle oorspronkelijke kenmerken van de tekst bewaard blijven: letterbeeld, eventuele beschadigingen, accenten. Soms moet ook de correspondentie tussen verschillende varianten van een tekst worden ingevoerd. Ook kan de filoloog kenmerken aan tekst-elementen willen meegeven, zoals een morfologische kodering, of het lemma. Vooral niet-romeinse schriftbeelden leveren problemen op. Afhankelijk van de computer waarmee men moet werken worden allerlei ad-hoc oplossingen gevonden. Was men in het begin afhankelijk van ponsband en ponskaart als invoer-medium, tegenwoordig worden teksten vaak ingebracht via beeldschermen, waarbij de tekst naar een magneetband of -schijf wordt overgebracht. Sinds kort zijn er programmeerbare beeldschermen op de markt waarbij men de vorm van de karakters zelf kan definiëren (binnenkort op de computer-afdeling aanwezig). Omdat vele drukkerijen zijn overgegaan op het gebruik van een computer voor het druk-klaar maken van teksten is het tegenwoordig steeds meer mogelijk om teksten direkt op een magneetband van een uitgever te betrekken. Wel moeten dan vaak voor de onderzoeker niet ter zake doende coderingen, zoals zet-instrukties, uit de tekst verwijderd worden. Als de tekst eenmaal in het computer-geheugen is gebracht zijn er op de meeste computer-systemen programma's voorhanden (zogenaamde "editor"'s) waarmee snel wijzigingen in de tekst kunnen worden aangebracht (eventueel ook systematische, zoals: verander alle "ph"'s in "f").
b. Uitvoer.
Bij de computer-uitvoer van bewerkingen op teksten speelt de afdruk-mogelijkheid een grote rol. Een eenvoudige line-printer kan alleen romeinse hoofdletters afdrukken. Bij wat duurdere typen is de zogenaamde print-ketting te vervangen door een andere, waarop bijvoorbeeld Gothische letters staan. Voor elk lettertype moet dan een speciale print-ketting gemaakt worden: een dure aangelegenheid. Een andere mogelijkheid bieden de zogenaamde printer-plotters: net zoals op bepaalde beeldschermen vormt men zelf de karakters door ze met puntjes samen te stellen. Ook zijn er enkele toepassingen bekend waarbij ingewikkelde tekens, zoals in het Chinees, door een plotter (een tekenmachine die door de computer bestuurd wordt) getekend worden. De techniek van het fotografisch zetten maakt het momenteel ook mogelijk om de computer-output direkt in boekvorm te laten afdrukken.

Lexicologie / Lexicografie / Lexicometrie.

Een oud handwerk van filologen is geweest het vervaardigen van indices en concordanties op door hen veel gelezen teksten. Hierbij wordt een gesorteerde lijst gemaakt van alle woorden die in de tekst voorkomen, met indicatie van hun vindplaats(en). Bij een concordantie wordt bovendien een deel van de omringende tekst van het woord getoond. Bij een gelemmatiseerde concordantie / index worden de woorden gegroepeerd onder het lemma dat zij gemeenschappelijk hebben. Een bijzonder probleem daarbij is de vorming van het lemma: voor slechts weinig talen is het tot nu toe gelukt om een algorithme te maken waarmee uit een tekst-woord een lemma wordt gevormd. Soms is daarbij de kennis benodigd die men ook voor het machinaal vertalen nodig heeft.
Het monnikenwerk van het vervaardigen van een index / concordantie is bij uitstek uit te besteden aan een computer. De kenmerken: snelheid en betrouwbaarheid zijn hier in het bijzonder van toepassing. Er zijn een groot aantal mensen geweest die hiervoor bijna gelijkluidende programma's hebben ontwikkeld, en er zijn heel wat boeken verponst met het uitsluitende doel er een concordantie of index van te maken. De stapel papier die de

computer daarbij produceert is indrukwekkend. Wanneer men geïnteresseerd is in combinaties van meerdere woorden verliest de index / concordantie echter haar waarde. Een aantrekkelijk alternatief is dan om de computer via een vraagtaal alle zinnen te laten geven die aan een bepaald patroon voldoen. Dit kan bijvoorbeeld een woordvolgorde betreffen. Een aantal grote computer-leveranciers brengen hiervoor programma's uit, vooral ontwikkeld voor het localiseren van korte beschrijvingen van publikaties die die bepaalde combinaties van woorden bevatten.
De laatste jaren komen steeds meer tekst-corpora beschikbaar die voorzien zijn van extra informatie, bijvoorbeeld achter elk woord een morfologische kodering, of het lemma. De genoemde documentatie-programma's zijn hier dan weer niet op ingesteld. Op onze computer-afdeling is een vraag-programma aanwezig dat ook deze situatie aankan (voor een beschrijving, zie hoofdstuk 5).
Bij de Lexicometrie worden allerhande tellingen gedaan aan woorden en woorddelen. Deze kunnen bijvoorbeeld een nuttige aanwijzing geven voor het verschil in stijl tussen schrijvers onderling. Ook wordt wel gepoogd om op deze manier een semantische maat voor tekst-woorden te kreëren. Statistische programmatuur is daarbij vaak vereist.
De computer speelt een hoofdrol bij het maken van "tekst-stambomen": een recent congres was gewijd aan methoden en algorithmen voor het zodanig classificeren van parallelle teksten dat een genealogie mogelijk wordt.

Grammatica's.

Een belangrijke aktiviteit van de taalkunde is het opstellen van regels om verschijnselen in een taal te beschrijven. Het ontleden is dan het herkennen van deze regels in een stuk tekst. In de informatica is eenzelfde aktiviteit waar te nemen. Alleen wordt de volgorde hier omgedraaid: men bestudeert de eigenschappen van formele methoden om kunst-talen te beschrijven en de eigenschappen van automaten om deze kunst-talen te ontleden. De laatste jaren is er groeiende belangstelling in algorithmen om een gegeven grammatica automatisch in een ontleder om te zetten die zijn werk doet in een zo kort mogelijke tijd.
Een vervolg-aktiviteit is om de computer automatisch een grammatica van een bepaald type te laten maken op basis van een aantal taaluitingen ("grammaticale inferentie"). Slechts voor een beperkt aantal typen kunst-talen zijn hier methoden voor ontwikkeld, en er is nog weinig theorie-vorming over de mogelijkheden en de complexiteit van inferentie-algorithmen voor andere typen. Dit alles geldt dan nog voor kunst-talen.
Mocht het mogelijk zijn om natuurlijke talen met dezelfde typen grammatica's te beschrijven als formele talen, dan liggen er interessante mogelijkheden in het verschiet. Er is echter nog steeds onderzoek gaande naar de beschrijvings-mogelijkheden van natuurlijke talen. Omdat deze bestaan uit rijen van tekens is een beschrijving in een formele taal mogelijk. Deze wordt echter onnodig ingewikkeld wanneer daarbij de betekenis van zins-delen niet betrokken wordt. Een probleem is ook dat deze beschrijving door taalkundigen moet worden gegeven, waarvoor het formalisme van kunst-talen niet flexibel genoeg blijkt te zijn, hoe men ook experimenteert met meer aansprekende technieken.
Ook het grote aantal grammatica-regels dat nodig is om een natuurlijke taal te beschrijven compliceert het probleem: men weet niet meer of een nieuw ingevoerde regel een oude niet tegenspreekt. Bij dit soort aktiviteiten kan de computer echter wel een helpende rol spelen: aan de hand van de opgegeven grammatica-regels en de interpretatie daarvan door een algorithme kunnen door de computer zinnen gemaakt worden. De onderzoeker controleert dan of de zinnen taalkundig goed zijn en stelt eventueel zijn grammatica weer bij. Op deze manier werkt een programma voor transformationele grammatica's (Joyce Friedman). Een soortgelijke aktiviteit vindt in onze faculteit plaats voor functionele grammatica's (zie de bijdrage van Kwee in hoofdstuk 4).
Een andere benadering is om de onderzoeker niet te dwingen volledig te zijn in zijn beschrijving maar hem toe te staan om, in herkennende zin, patronen op te geven waaraan een zin moet voldoen. Dit kan dan gecontroleerd worden aan de hand van een corpus met correcte zinnen. Op de computer-afdeling wordt hieraan onderzoek gedaan.

Semantiek.

Uit de vorige paragraaf bleek dat formele grammatica's niet adekwaat zijn om een natuurlijke taal te beschrijven wanneer er geen voorziening voor semantische regels wordt aangebracht. Dit was in feite het breekpunt bij het onderzoek naar automatische vertaling. Voor de disambiguering van een bepaald woord kan het nodig zijn om de betekenis van omringende woorden te kennen. Er is daarom de laatste 10 jaar op kleine schaal geëxperimenteerd met methoden om de betekenis in de computer te kunnen noteren en om deze te kunnen gebruiken bij het herkennen van zinnen, paragrafen en zelfs verhalen die dan over een zeer beperkt onderwerp moeten gaan. De ontleding werkt dan met een mengelmoes van syntactische en semantische patronen.
Methoden om de betekenis van een woord te kunnen opbouwen uit een groot aantal teksten zijn nog nauwelijks ontwikkeld. Om een schaalvergroting te krijgen wordt ook gestudeerd op mogelijkheden om semantische lexicons aan te leggen. Met wat fantasie kan men zich vele experimenten voorstellen waarbij ten behoeve van inferentie gebruik gemaakt wordt van tekst-corpora die van toevoegingen zijn voorzien. De wijze van opslag binnen de computer speelt hierbij een vitale rol. Aan het laatste wordt op de computer-afdeling onderzoek verricht.
In het algemeen spelen technische mogelijkheden binnen de computer-taalkunde een veel

grotere rol dan uit publikatie blijkt: projekten worden hierdoor vaak gestart (door de uitdagende aanwezigheid van een computer) of juist op de lange baan geschoven (vanwege de beperkingen van die computer). De frustratie door gebrek aan computer-tijd, te complex geworden programma's, te weinig computer-capaciteit of te weinig mankracht laten per onderzoeker het onderzoeks-gebied vaak verschuiven.

Taalkunde.

Toepassingen van tot nu toe besproken technieken zijn binnen de taalkunde te vinden. Wanneer deze verder zullen zijn ontwikkeld (bijvoorbeeld die van de semantische voorstellingen) zullen zij bijvoorbeeld toegepast kunnen worden in tekst-grammatica's ("discourse analysis"). Statististische methoden worden veelvuldiger toegepast bij vergelijkend onderzoek naar taaluitingen van bijzondere groepen (kinderen, moeder-kind, geslacht, buitenlanders, sociale groepering, pathologie, historisch). Hierbij kunnen allerlei computer-hulpmiddelen gemobiliseerd worden, zoals het grafisch vertonen of plotten van de resultaten. Statische programmatuur heeft voldoende rijpheid, betrouwbaarheid en toegankelijkheid gekregen om voor deze toepassingen met vrucht gebruikt te worden. (Bijvoorbeeld het pakket SPSS).

Taalverwerving.

Statische methoden kunnen eveneens toegepast worden bij het vergelijken en behandelen van lesmateriaal ("wanneer wordt een woord voor het eerst geintroduceerd", "hoeveel lastige woorden staan er naast elkaar", enzovoorts). Voor het geven van lessen dient de computer zich aan als instrukteur. Een experiment daarmee aan de UvA is de laatste twee jaar gedaan met het "Plato-systeem". De gedachte achter dit systeem is om de leraar te ontlasten van routine-werk. De computer kan steeds betere onderwijs- strategieën leren waardoor de leerling een efficiënte terugkoppeling krijgt op zijn vorderingen en zijn zwakke punten, onafhankelijk van de persoonlijke kwaliteiten van een leraar en de grootte van zijn studiegroep.

Documentatie en archivering.

Ten behoeve van archivering, catalogisering en uitlening in bibliotheken zijn een groot aantal systemen ontwikkeld, vaak toegespitst op de gebruikte computer en randapparatuur. Commerciële leveranciers bieden momenteel hun diensten aan voor het raadplegen van enorme catalogi door middel van het opgeven van combinaties van index-woorden.
In de laatste jaren zijn er ook terminologische data-banken ontstaan die dienen als multi-lingual woorden boeken (voorbeeld: de Canadese regering heeft een dergelijke data-bank service met 70 aangesloten terminals waartoe ongeveer 800 vertalers toegang hebben). Bij een aantal documentatie-systemen is sprake van het toevoegen van een taalkundige component. Is een zoek-term niet aanwezig dan wordt gezocht naar bijvoorbeeld een synoniem, of een hiërarchisch woord, of een woord dat een (groot) aantal letters gemeenschappelijk heeft, of dat dezelfde stam heeft.

Archeologie.

Bij de verwerking van grote vondsten wordt de computer met succes gebruikt. Van voorwerpen (bijvoorbeeld aardewerk) worden verschillende kenmerken (zoals versieringen, manieren van bakken, randen, vindplaats) geregistreerd. Met behulp van statistische technieken (zoals cluster-analyse) probeert men dan de voorwerpen in bepaalde groepen in te delen.
Aan de hand van de vindplaatsen kunnen door de computer verspreidings-kaarten worden gemaakt. Op deze manier is door archeologen van onze faculteit gewerkt bij opgravingen te Assendelft.

Geschiedenis.

Door het aanleggen van vrij gedetailleerde data-bestanden voor een beperkt gebied over een aantal jaren hoopt men inzicht te krijgen in het verloop van een bepaald verschijnsel. De gegevens betreffen bijvoorbeeld namen - leeftijden - beroepen - woonplaats - inkomen - bezittingen - enzovoorts van mensen (Philadelphia projekt), schepen die langs een bepaalde plaats voeren: data - bestemmingen - ladingen - enzovoorts (Sont projekt), theater - plaats - tijd - stuk - spelers - regie - enzovoorts (London Stage projekt).
Voor de meeste van deze projekten geldt dat zij zonder veel financiële steun zijn opgezet en dat programmatuur zelf ontworpen moest worden. Ook nu nog zijn er geen goede documentatie- / data-base systemen die de veelheid van relaties die in deze gegevens aanwezig is behoorlijk aankunnen.

Musicologie.

Het genereren van muziek door een computer is een spectaculair gebeuren. Lastiger is het analyseren van muziek en het herkennen van bijvoorbeeld componist, tempo-verschillen en instrumenten. De signaal-analyse en patroon-herkenning die daarvoor nodig zijn vereisen een brede technische- / wiskundige oriëntering.

Fonetiek.

In de fonetische wetenschappen is het proces van spreken en verstaan onderwerp van onderzoek. Het doel is te komen tot een theorie die de interaktie tussen spreker en luisteraar op basis van spraak beschrijft. Om dit te bereiken worden deelprocessen zowel afzonderlijk als in onderlinge samenhang bestudeerd. Het is nu mogelijk een aantal van deze deelprocessen modelmatig te beschrijven met exacte hulpmiddelen. Dit kan leiden tot modelvoorstellingen die een mathematisch-fysisch, stochastisch, signaal-analytisch dan wel statistisch karakter bezitten. Om eigenschappen van een model zichtbaar te maken is het soms nuttig, soms noodzakelijk, gebruik te maken van een rekenmachine. Voorbeelden van onderzoek waarbij dit plaatsvindt zijn:

- De reconstructie van de vorm van de mondkeelholte op grond van het spraaksignaal.
- Wiskundige analyse / synthese van het spraaksignaal; het zichtbaar / hoorbaar maken van de resultaten.
- Onderzoek naar de transformaties die het spraaksignaal in het binnenoor ondergaat.
- Onderzoek naar methoden waarmee de uitspraak van een spreker kan worden gekarakteriseerd.
- Verwerking van gegevens zowel afkomstig van metingen aan het spraaksignaal als van perceptie-experimenten.

In hoofdstuk 4 worden de eerste twee voorbeelden nader toegelicht onder het hoofdje 'spraaksignaal analyse'.

Hoofdstuk 4. Wat is er zo al gedaan in onze faculteit ?

De afgelopen jaren hebben onderzoekers en studenten uit de volgende vakgroepen gebruik gemaakt van de diensten van de computer-afdeling:

Subfaculteit I:
- Nederlandse Taalkunde
- Historische Nederlandse Letterkunde en Tekstinterpretatie

Subfaculteit II:
- Duitse Taal- en Letterkunde
- Engelse ,,
- Franse ,,
- Roemeense ,,
- Spaanse ,,
- Italiaanse ,,
- Slavische ,,

Subfaculteit III:
- Oudgriekse ,,
- Latijnse ,,
- Westaziatische Talen en Culturen

Subfaculteit IV:
- Nieuwe Geschiedenis
- Economische- en Sociale Geschiedenis

Subfaculteit V:
- Algemene Taalwetenschap
- Fonetische Wetenschappen
- Muziekwetenschap

Voor zover gebruik gemaakt kon worden van reeds ontwikkelde programmatuur werd ook incidenteel met onderzoekers van buiten de faculteit samengewerkt (mits door hen een binding met de faculteit kon worden aangetoond). Hieronder vielen:

- het Instituut voor Dialectologie te Amsterdam
- de vakgroep Germaanse talen van de UCLA
- de vakgroep Nederlandse taal- en letterkunde van de universiteit van Metz
- de vakgroep Talen van de TH-Eindhoven
- de vakgroep Algemene Taalwetenschap van de RU
- de vakgroepen Fonetische Wetenschappen te Groningen, Utrecht en Nijmegen
- enkele promovendi van de VU.
- de interfaculteit Bedrijfskunde van de TH-Delft
- een werkgroep van gastdocent Findler aan de UvA (kunstmatige intelligentie).

Het gebruik varieerde van eenvoudig sorteerwerk tot langdurige (meerdere jaren beslaande) projekten.
Met enkele van die grotere projekten laten wij u nu kennis maken. Hierbij laten we de onderzoekers zelf aan het woord. Soms zal kort worden aangegeven wat de bijdrage van de computer-afdeling aan deze projekten was. In hoofdstuk 5 worden sommige aspekten daarvan verder uitgewerkt. Ook zullen enkele onderzoekers aan het woord komen die grotendeels zelfstandig gebruik maken van het computer-systeem.

Het onderzoek naar de Materiële Cultuur in de Nederlandse steden (1500-1800)

Dit onderzoeks-projekt beoogt meer te weten te komen over de woon- en leefgewoonten van onze voorouders in Nederland. Met behulp van boedelinventarissen die op grote schaal in het verleden opgemaakt zijn, proberen wij een beter inzicht te krijgen in de sociaal breed gespreide rijkdom in de Republiek en wordt er onderzocht of er na het tijdperk van de 'Gouden Eeuw' sprake is van een toenemende verarming en verpaupering van grote delen van de bevolking.

Ook modeverschijnselen en cultuuroverdracht, wat betreft de materiële goederen, kunnen uit de boedelinventarissen gedestilleerd worden, terwijl voor de economische geschiedenis van ons land een onderzoek naar de ontwikkeling van de consumptieve vraag naar duurzame gebruiks-goederen van groot belang is.

Omdat in de boedelinventarissen zowel de roerende als de onroerende goederen per gezin beschreven staan, kan men ook een idee krijgen over de wijze waarop de mensen vroeger hun geld belegden, en of hierin ook verschuivingen en veranderingen te bespeuren zijn.

Mijn taak in dit onderzoeks-projekt is om met een betrekkelijk kleine stad in Nederland (Delft) een begin met het onderzoek te maken en om met het door de computer-deskundigen van de computer-afdeling te ontwerpen computer-programma een systeem te ontwikkelen om de bronnen, in casu de boedelinventarissen, zodanig te kunnen bewerken dat er later op veel grotere schaal onderzoek in vele Nederlandse steden - ook in Amsterdam - verricht kan worden, waardoor we een soort 'materiële' kaart en inventaris van het verleden kunnen maken, en de verschillen van de diverse steden en regio's zowel wat betreft leef- en woon-gewoonten als wat betreft spreiding van welvaart en modeverschijnselen kunnen aantonen.

Het onderzoek naar de materiële cultuur is een nieuwe richting in de geschiedwetenschap die op het ogenblik op vele plaatsen in Europa van de grond aan het komen is. Voor zover wij weten, zijn wij hier in Nederland de eersten die met behulp van de computer op dit terrein werkzaam zijn, wat opmerkelijk is als men bedenkt dat in elke boedelinventaris zeer veel verschillende voorwerpen en gegevens voorkomen die tot enkele duizendtallen kunnen oplopen. Om tot enige redelijke en plausibele conclusies te kunnen komen zal er toch met zo'n 300 a 400 boedelinventarissen gewerkt moeten worden per stad of distrikt, zodat het begrijpelijk is dat het bewerken van al dit bronnenmateriaal met de hand een onmogelijke zaak is.

Th.F. Wijzenbeek-Olthuis

Bijdrage van de computer-afdeling:
- het opstellen van een bestands-grammatica
- noodzakelijke assistentie bij data-invoer en data-beheer.

Het filologische handwerk en de computer.

Als je als filoloog eenmaal met de computer begint te werken, wil je er ontzettend veel mee doen. Dat kan ook! Het blijkt in de praktijk dat je het meeste plezier beleeft aan de hulp die je geboden wordt bij een aantal zeer weinig spectaculaire bezigheden, die echter zonder de computer tijdrovend, vervelend en zelfs ondoenlijk zouden zijn. Dat er nog veel meer met de computer kan, weet ik ook wel, maar daar heb ik het nu niet over. Natuurlijk doe ik het wel. Het werk van een filoloog is voornamelijk van interpretatieve aard; in engere zin (het verklaren van de betekenis van woord of zin) zowel als in de bredere betekenis (het plaatsen van een tekst in de socio-culturele kontekst). Bij dat voortdurend werken met teksten, komt telkens weer als voorbereidende fase een enorme hoeveelheid 'noeste' arbeid! Er zijn in de geschiedenis heel wat filologen aan te wijzen waarvan de tijd en de aandacht zo zeer zijn opgegaan aan de ambachtelijke aspecten van hun werk, dat zij niet of nauwelijks aan interpretatie zijn toegekomen. Hier beschrijf ik uitsluitend hoe er met de hulp van de mensen achter de computer werktuigen zijn ontstaan die een gedeelte van het filologisch handwerk verlichten.

Ik bereid, met anderen, de uitgave voor van meer dan honderd Oud-Franse verhalen: fabliaux, gemiddeld 300 verzen lang (de kortste is 30 verzen, de langste 1200 maar er is een duidelijke tendens naar het midden).

De teksten zijn bewaard gebleven in middeleeuwse handschriften, en vaak is een verhaal opgetekend in meerdere handschriften die onderling veel of weinig verschillen.

Ik geef een voorbeeld, nl. een tekst waarbij de computer het werk zeer veel heeft geholpen: Auberee. Deze tekst is bewaard gebleven in zeven mss. die ik aanduid als A B C D E F J. De totale lengte is 600 verzen, sommige versies zijn korter, andere langer.

Het eerste werk dat gedaan moet worden is de teksten afschrijven naar de mss. Hier helpt geen computer aan! Als na controle door vijf(!) mensen we het erover eens zijn dat onze transcriptie weergeeft wat er in de 13e eeuw is opgeschreven, wordt de tekst verponst. Dit gebeurt door een afgestudeerd medewerkster Frans - Mei Khoe - die dit werk zo nauwkeurig doet dat een geschoold filoloog de grootste moeite heeft fouten te ontdekken. Zij doet het ook zeer snel, zodat we nu reeds beschikken over een bestand van 28000 versregels die de machine zo kan lezen. Op elke ponskaart staat telkens alleen maar een vers (v. 1 van Auberee naar hs. A)

QUI PRES DE MOI SE VOUDRA TRERE

Het kan niet genoeg benadrukt worden hoe eenvoudig het voor de gebruiker is. Hij gaat achter een schrijfmachine zitten die als voornaamste bijzonderheid heeft dat je niet na elk vers op een knop "nieuwe regel" drukt, maar op een knop "nieuwe kaart".

Als je nu het hele handschrift A hebt verponst (606 verzen = 606 kaarten) moet je nog een kaart maken die je boven op het stapeltje legt. Daar zet je op:

* FABLIAU 4 (A)

De machine zorgt er dan wel voor dat alle kaarten die erna komen worden beschouwd als behorende tot Auberee (= fabliau 4 van onze publikatie) naar hs. A. Datzelfde gebeurt ook met de andere hss van Auberee. De kaarten worden nu in een kaartleesmachine gestopt, op een andere schrijfmachine tik je nu een stuk of dertig regels uit met bevelen aan de machine. Wat je moet uittikken staat al keurig voor je opgeschreven op een blaadje. Alleen af en toe tik je de naam van jouw onderzoek in bijvoorbeeld Auberee of Fabliaux. Wanneer dat moet, dat geeft de machine wel aan doordat er op het beeldscherm vragen verschijnen als: "hoe heet je onderzoek?" Als je dan vijf minuten wacht krijg je er keurig uit wat je er in hebt gestopt, n.l. je eigen tekst maar nu gedrukt! En wel als volgt:

1. op elke regel staat een vers.
2. voor elk vers staat aangegeven uit welke fabliau en uit welk handschrift het vers afkomstig is.
3. alle verzen hebben een nummer gekregen. Dit is het nummer dat het vers ook werkelijk in de fabliau heeft: vers 353 van Auberee hs A heeft ook als nummer 353 gekregen!
4. Maar het allerbelangrijkste: alle woorden van de verzen staan in alfabetische volgorde; m.a.w. ik heb een belangrijk filologisch werktuig gekregen. Elk vers wordt helemaal afgedrukt terwijl het woord waar het om gaat voor de duidelijkheid KAPITAAL gedrukt is:

MANUSCRIPT 4 VAN HANDSCHRIFT A / 20 OKTOBER 1980 PAG 3

* FABLIAU 4 (A)	353	-et- tout souef a li ADOISE	353
* FABLIAU 4 (A)	292	a ta-n-t la borgoise s'ADRECE	292
* FABLIAU 4 (A)	580	n'ot mes tel ioie en son AEL	580
* FABLIAU 4 (A)	9	-et- riches -et- de gra-n-t AFERE	9
* FABLIAU 4 (A)	33	se li a son AFERE dit	33
* FABLIAU 4 (A)	38	ele n'est pas de ton AFERE	38
* FABLIAU 4 (A)	536	por recoudre -et- por AFETIER	536

5. Er achteraan komt nog een tweede werkinstrument: een lijst van dezelfde woorden, nu

zonder kontekst maar voorzien van hun frekwentie.

MANUSCRIPT 4 VAN HANDSCHRIFT A / 20 OKTOBER 1980
PAG 1

10	adoise	1
11	adrece	1
12	ael	1
13	afere	3
14	afetier	1

Nu kan de filoloog aan het werk. Enkele voorbeelden van wat ik ermee doe:

1. de middeleeuwse copiisten hebben de gewoonte een aantal woorden in afgekorte vorm op te schrijven. Wij schrijven ze voluit, maar we willen dit doen volgens de volgende regels:

a) Nagaan of het afgekorte element in dezelfde tekst in hetzelfde handschrift in hetzelfde woord of in een tot hetzelfde paradigma behorende woord ook in niet afgekorte vorm voorkomt. Indien dit het geval is en er maar een spelling is, wordt de afkorting volgens deze spelling opgelost.
b) In andere gevallen geldt een lijst van oplossingen waarbij etymologische of assimilatieregels worden toegepast.

Nu is dat nagaan van het voorkomen van niet afgekorte vormen een heel werk, als je het met de hand doet. In de concordantie-lijst, waarin de opgeloste afkortingen tussen de liggende streepjes staan, is dat echter snel gebeurd, en zo halen we er nog menige fout uit; cf. Auberee pag. 117 F 388 e-m-bracha moet gecorrigeerd worden in e-n-bracha zoals blijkt uit het niet voorkomen van embracha en het wel voorkomen van enbracha F 31 (pag. 127).

2. Bij het maken van het kommentaar merk ik dat hs A in vers 216 als enige tantost heeft, terwijl alle anderen maintenant schrijven. Nu betekent tantost 'dadelijk', en dat is ook de normale betekenis van maintenant in het Oud-Frans. Dit laatste woord moet echter langzamerhand de moderne betekenis van 'nu' hebben gekregen. Het lijkt wel alsof hs A maintenant = dadelijk niet meer kent. Het overzicht van alle gevallen waarin alle hss maintenant gebruiken bevestigt deze veronderstelling.
3. Er bestaat een hypothese dat veel van de verschillen tussen versies van de fabliaux verklaard kunnen worden uit het verschil in publiek; ze worden als het ware op het publiek toegeschreven rekening houdend met hun opvattingen.
Nu zijn er verschillen die in dit verband betekenisvol kunnen zijn, maar die slechts met behulp van concordantie- en frekwentie-lijsten te vinden zijn. Bijvoorbeeld de aanduiding van de hoofdpersonen. De bedrogen echtgenoot wordt als volgt aangeduid:

bourgeois	A B C D E F J
sire	A B C D E F J
baron	F
vilain	A B C D E J
mari	D E F J

Uit de lijsten blijkt dat het gaat om twee passages waarin een andere hoofdpersoon, de koppelaarster, deze man aanduidt met vilain. Het is dan duidelijk dat F ten behoeve van zijn publiek dit scheldwoord opgevat heeft als een verkeerde klasse-aanduiding, wellicht zelfs als beledigend voor de klasse van "bourgeois" waartoe de man blijkens meer dan tien passages gerekend wordt, en dit heeft veranderd in bourgeois (F 135) en barons (F 303). Dit laatste woord komt uitsluitend in F voor!

Natuurlijk kun je tegenwerpingen maken:

a) Dat kun je toch allemaal met de hand doen! Maar natuurlijk! Alleen..... om te weten dat een woord maar één keer en dan wel alléén in F staat, moet ik ruim 7 x 600 = 4200 verzen doorlezen. Nu doet de machine dat voor me!
b) Je krijgt van die enorme pakken papier waartussen ook heel veel zit waar je niet naar kijkt. ook dit is waar. Er is echter ook een programma Query dat het mogelijk maakt de antwoorden op vragen als: geef alle passages waarin bourgeois staat eerst op het beeldscherm. Als je het antwoord interessant vindt kun je het ook nog gedrukt krijgen!
c) Een filoloog hoort zijn teksten te kennen 'van binnen uit' door ze goed en grondig te lezen! Dit is juist en heeft dan ook met het al of niet gebruiken van de computer niets te maken! Hij moet zijn teksten blijven lezen en bestuderen zoals hij altijd heeft gedaan. Maar de beste filoloog kan niet wat de domste computer wel kan: onthouden wat er in een tekst niet heeft gestaan.

En de computer kan nog veel meer! Er wordt dan ook gewerkt aan het automatisch vergelijken van de varianten in handschriften.

H.J. v.d. Boogaard

Bijdrage van de computer-afdeling:
- teksten geschikt maken om interaktief naar patronen te kunnen zoeken
- routine-matig draaien van een concordantie-programma.

Het gebruik van de computer voor Hettitische teksten in Spijkerschrift.

Het gebruik van de faciliteiten van de facultaire computer door de vakgroep Westaziatische talen en culturen is een voortzetting van vroegere aktiviteiten, waarbij elders concordanties op lemmata en grammaticale en syntactische classificaties van twee tekst-corpora van Oud- en Middel-Hettitische teksten uit respectievelijk 1700-1500 en 1450-1380 v. Chr. werden vervaardigd. Thans wordt getracht de computer-verwerking zo te laten verlopen dat een deel van het ingevoerde materiaal tevens als tekst-editie kan worden gepresenteerd. Op practische gronden werd besloten een vrij omvangrijke groep van Oud-Hettische historische teksten uit het tijdvak van ca. 1630-1595 v. Chr., een goed te dateren subgroep dus van het Oud-Hettitische materiaal, dat reeds eerder werd verwerkt, met dit doel in te voeren. Terwijl bij de gebruikelijke tekst-edities slechts afwijkingen uit de duplicaten in een varianten-apparaat plegen te worden genoteerd, is de invoer bij dit projekt zo geconstrueerd dat de computer ook bijhoudt, in hoeverre de secundaire "manuscripten" het voor de tekst-editie preferent verklaarde kleitablet ondersteunen. Op dit moment werken

1) het programma, dat de ingebrachte tekst naar de samenstellende "manuscripten", die dus integraal afgedrukt worden, uitsplitst, en
2) het aansluitende programma, dat vervolgens een uitvoer produceert, die een vergelijking met een normale, met de hand gezette tekst-editie kan doorstaan, reeds zeer bevredigend. Het merendeel van de voetnoten wordt hierbij automatisch gegenereerd.

In de toekomst zal de programmatuur, die thans bij de medewerkers van de computer-afdeling in bewerking is, verder uit de volgende onderdelen bestaan:

3) een semi-automatisch lemmatiserings-programma;
4) een programma om interaktief naar lemmata, grammaticale en syntactische codes en schriftconfiguraties te zoeken;
5) programma's voor het aanmaken van index-files, de sortering daarvan en de afdruk in gespecificeerde index-vorm.

In een reeks van besprekingen met Van der Steen werd met het oog op deze programmatuur een invoer-notatie ontwikkeld, die het de computer mogelijk moet maken de lemmatisering voor een groot gedeelte voor eigen rekening te nemen, terwijl de grammaticale codes zo werden gedefinieerd dat deze later tevens gebruikt zullen kunnen worden om bij de index-vorming de binnen een lemma gewenste onderverdeling van het materiaal aan te brengen. Het interaktieve zoek-programma zal bijzonder nuttig zijn voor het onderzoek naar het gebezigde Syllabair (het totaal van de syllabische waarden in hun basis-variatie CV-VC versus CVC zowel als in hun contrast ten opzichte van een alternatieve weergave met woordtekens) en voor spellingskwesties. Voorts zal dit programma zeker een nuttige rol spelen bij het controleren, of de grammaticale en syntactische codes consistent zijn toegekend.

Zoals uit het bovenstaande moge blijken, wordt getracht om de computer-verwerking zo intensief mogelijk te exploiteren. Degene, die de invoer verzorgt, fungeert immmers tezelfdertijd als "zetter" voor dat onderdeel van het corpus, dat voor een tekst-editie bestemd kan worden. De gekozen opzet van het projekt is namelijk mede ingegeven door de vrees dat de conventionele verwerking van Hettitologische publikaties binnen afzienbare tijd illusoir zal zijn. Het ligt in de bedoeling om de reeds ontwikkelde en de nog te vervolmaken programmatuur voor enkele tekst-edities met indices van historische en religieuse Hettitische teksten te gebruiken en tegelijkertijd een tamelijk omvangrijk corpus van Nieuw-Hettitische teksten (1345-1200 v. Chr), op te bouwen, dat met de reeds beschikbare corpora uit de oudere periodes kan worden gecontrasteerd. Na de boven reeds genoemde Oud-Hettitische historische teksten zullen in het komende kalenderjaar de nog niet eerder als geheel bestudeerde groep van religieuse "Gelofte-teksten" uit de dertiende eeuw v. Chr. worden ingevoerd. Op deze tekst-editie, die in 1981 zal dienen te verschijnen, zal, naar wij hopen, in principe het geheel van de programmatuur toegepast kunnen worden.

Ph.H.J. Houwink ten Cate

De bijdragen van de computer-afdeling zijn hier boven al aangeduid. Hieronder volgt een voorbeeld. De invoer weerspiegelt de specialistische kennis van de filoloog terwijl de uitvoer de publicatievorm van de hedendaagse Hettitologen benadert. We zullen daarom op 3 regels van de in- en uitvoer een lopend commentaar geven zodat een globaal idee verkregen kan worden van de mogelijkheden.
Het voorbeeld betreft een tekst die op 6 verschillende kleitabletten (elk op andere plaatsen afgebroken en beschreven) is gevonden. Voor de eenvoud lichten we er een deel uit dat slechts op 2 tabletten voorkomt (die we zullen noemen manuscripten A en B). De uitvoer betreft de 2 manuscripten afzonderlijk, waarbij afbrekingen worden gerestaureerd op basis van de tekst van het andere manuscript (tussen ronde haken) of worden aangevuld door de filoloog zelf (tussen rechte haken). Als de manuscripten in tekst van elkaar verschillen (varianten) dan wordt dit door de computer in een voetnoot gemeld.

+A3' +B2' @ [ERÍN.MEŠ]⌣AB, #0BCY# *Ù* #14CKB# GIŠGIGIR/VB,

I.MEŠ,BV #0BDY# @ #0BCY# ,B⌣ *A-NA*=A, In this last sign

KBo 22-4 3' (right) joins to KUB 40-5 II 1' (left). Line-

numbering of KUB 40-5 (II) 1' henceforward adapted. ,A=

#10CB# LÚ #0BAY# URU*Za-al-pa/* #17AGA# ,A⌣[(8)]]

+A4' +B3' [*p**e*\$*i*$-*r*]⌣AB,*a-an*%$*n*$*a* #10AAA# #14BAD#

M*At-ra-d*,B⌣*u/-uš* #17ADB# *þu-u-ja-an-z*,A⌣[*a*]

{*þuyāi-/þuija-*} #3AAAD#

Commentaar op de 3 regels die beginnen bij +A3' :

+A3'	Voor manuscript A begint de derde regel van boven (bovenkant van tablet afgebroken). Het begin van de regel is afgebroken.
@	Begin van een aantal woorden die ook gezamenlijk als een geheel in de index moeten worden opgenomen.
[	De filoloog vermoedt dat hier de volgende tekst heeft gestaan.
ERÍN.MEŠ	Een sumerisch woord.
]	Einde van het vermoeden.
⌣AB,	Bij zowel manuscript A als B komt de tekst beschikbaar.
#0BCY#	Het voorgaande woord krijgt de morfologische klassificatie 0BCY (dit kan later in een zoekpatroon betrokken worden).
Ù	Akkadisch woord
GIŠGIGIR	Sumerisch woorddeel met een voorvoegsel (bij elkaar 2 spijkertekens)
/	Voor het bepalen van het lemma kan het hele woord genomen worden.
VB,	Manuscript B gaat afwijken van A.
.MEŠ	Restant van het vorige Sumerische woord.
,BV	Einde afwijking van B. Er volgt geen afwijking voor A, dus B had het vorige woorddeel extra (in de uitvoer wordt hier een voetnoot gegenereerd).
@	Einde van het eerder aangeduide woord-geheel.
,B⌣	Manuscript B breekt af.
A-NA	Akkadisch woord
=A,	De filoloog gaat nu commentaar leveren op alléén manuscript A.

Een stukje uitvoer voor manuscript A is (tezamen met voetnoten) :

1' [ ]x[x]x[

2' [(-⌜*aḫ*⌝-⌜*ḫi*⌝-⌜*iš*⌝)] ⌜LUGAL⌝-*i ku-ru-r*[*i-ja-aḫ*-.....

3' [ERÍN.MEŠ] *Ù* GIŠGIGIR[8]) *A-NA*[9]) LÚ URU*Za-al-pa* [

4' [*pí-r*]*a-an-na* M*At-ra-du-uš ḫu-u-ja-an-z*[*a*]

8) B II? 2' .MEŠ ADDIDIT

9) In this last sign KBo 22-4 3' (right) joins to KUB 40-5 II 1' (left). Line-numbering of KUB 40-5 (II) 1' henceforward adapted.

regel 1': x A begint met aan de bovenkant 2 onleesbare tekens (x).
2': Eerste deel van deze regel afgebroken, maar ten dele te restaureren op basis van B. Wordt aan de rechterkant onleesbaar. De filoloog meent dat hier nog i-ja-ah- gestaan kan hebben, maar tast over het vervolg in het duister (.....).
⌜ah⌝ gedeeltelijk beschadigd teken.
3': Zie het commentaar bij de invoer. Voetnoot 8 wordt automatisch gemaakt, voetnoot 9 is ingevoerd.
[Aan de rechterkant is de regel afgebroken en het is niet te bepalen hoeveel tekst daar nog gestaan heeft.

etc.

Hieronder volgt hetzelfde tekstdeel voor B (hierop hebben geen voetnoten betrekking):

1' [ ]-ʳ*ab*¬-ʳ*bi*¬-ʳ*iš*¬ [(ʳLUGAL¬-*i ku-ru-r*)*i-ja-ab-*.....

2' [ERÍN.MEŠ] *Ù* ᵍᶦˢGIGIR.MEŠ [(*A-NA* LÚ ᵘʳᵘ*Za-al-pa*)

3' [*pí-r*]*a-an-na* ᴹ*At-ra-d*[(*u-uš bu-u-ja-an-z*)*a*]

Dit behoeft geen nieuw commentaar.
Een gerestaureerd kleitablet (op basis van de manuscripten A en B) zou dan de volgende tekst bevatten :

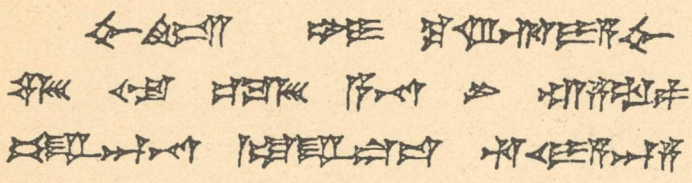


Fig. 1

Een gelemmatiseerde concordantie van "Die Schweizer Minnesänger."

De computer-assisted research op het gebied van het middeleeuwse Duits omvat meer dan 100 titels. Opvallend is dat de meeste publikaties meer plannen bevatten dan werkelijk uitgevoerd onderzoek. Alleen het maken van indices en concordanties is vrij veel gebeurd. Bij de gelemmatiseerde concordanties die tot nog toe verschenen zijn, is men altijd als volgt te werk gegaan: men heeft de woordvormen van de te lemmatiseren tekst van te voren manueel van codes voor lemma's, woordsoort, declinatie, conjugatie, enz. voorzien, waarna dit bewerkte materiaal door de computer geordend is.
Al spoedig zocht men wegen om het lemmatiseren door de computer te laten verrichten. Twee methodes zijn daarvan bekend. De eerste is, dat men, uitgaande van een concordantie die, zoals boven beschreven, manueel gelemmatiseerd is (het Arbeitswörterbuch), andere corpora automatisch wilde lemmatiseren. De computer moest dan alle woordvormen van het niet-bewerkte nieuwe corpus vergelijken met het Arbeitswörterbuch en bij voorkomen de gegevens van het Arbeitswörterbuch bij de woordvormen van de te maken concordantie vermelden. De moeilijkheden bleken onoverkomelijk. Immers:
1 de vele woordvormen uit het nieuwe corpus die niet in het Arbeitswörterbuch voorkomen, moeten weer manueel bewerkt worden,
2 de vele homografen moeten manueel gescheiden worden,
3 de niet-uniforme orthografie van het oudere Duits moet eerst manueel genormaliseerd worden.
Deze methode waarmee men in Bonn jarenlang gewerkt heeft en waarin de Deutsche Forschungsgemeinschaft zeer veel geld gestoken heeft, is op een mislukking uitgelopen.
De tweede methode is wetenschappelijk veel ambitieuzer. Men probeert door het formuleren van regels de lemma's, woordsoorten, declinaties, conjugaties, enz. in "Flexionsalgorithmen" voor de computer herkenbaar te maken. De weinige pogingen die voor een enkele woordsoort ondernomen zijn, stemmen niet erg hoopvol. De oorzaak is principieel: onze taalkundige kennis van het oudere Duits schiet te kort om het voor de computer noodzakelijke regelsysteem te formuleren.
Wijs geworden door bovenstaande ervaringen hebben wij een eenvoudige, maar wel werk-intensieve methode van lemmatiseren gekozen. Ons projekt is internationaal aangemeld en aangekondigd en houdt in, het vervaardigen en verantwoorden van een volledige gelemmatiseerde concordantie op de Zwitserse lyriek van de 13e eeuw (tekst-uitgave: Karl Bartsch (Hg)., Die Schweizer Minnesänger, Frauenfeld 1886, Darmstadt 1964). Afwijkend van de tot nog toe gebruikte methodes hebben we de computer de onbewerkte tekst eerst laten alfabetiseren om overeenkomstige vormen zoveel mogelijk bij elkaar te hebben. Daarna is met diverse codes de hele tekst manueel gelemmatiseerd. Dit was een interpretatorisch en taalkundig vaak zeer moeilijk werk.
Op de volgende pagina ziet u een fragment van de definitieve versie.
Bij de definitieve versie van de concordantie stond vooral het gezichtspunt van overzichtelijkheid centraal. Wij hebben dan ook zo min mogelijk gebruik gemaakt van toegevoegde gegevens. Alleen daar, waar het absoluut noodzakelijk was het opzoeken van bepaalde gegevens voor de gebruiker te vergemakkelijken, hebben we toevoegingen gemaakt. Homografen zijn bijvoorbeeld van een woordsoort-aanduiding voorzien (herte als adjectief en herte als substantief). Kruisverwijzingen zijn aangebracht bij bijvoorbeeld adjectieven en adverbia, die door klankwisseling niet onder hetzelfde lemma genoemd zijn (herte en harte). Veel woorden, die nu eens als simplex dan weer als composituur in het te bewerken materiaal optraden, zijn onder het composituur ondergebracht, terwijl via kruisverwijzingen bij het simplex hierop attent gemaakt wordt (herze).
Bovendien is in de concordantie een "Reimwortverzeichnis" opgesloten. Hiermee in verband staan de sterretjes en plusjes in de tekst.
Meestal waren de laatste woorden van de regel in deze lyrische teksten rijmwoorden. Deze hoefden niet apart gemarkeerd te worden. Binnenrijmwoorden en rijmloze woorden, op plaatsen waar men een rijmwoord verwachtte, zijn voorzien van een sterretje respectievelijk plusje.

HERTE /HERT *‹adj.›* s.a. harte
  HERT
    WIN, 2, 8 ouwê die sint alsô HERT
    HAD, 2, 9 doch was sî gar* HERT wider mich.
  HERTE
    BUW, 1, 22 waer mîn herze ein adamas sô HERTE,
  HERTER
    MHT, 2, 11 Ach, HERTER muot*, waz hât dîn diu güetlîch gestalte zühte rîche
    ROS, 8, 9 al mîn nôt mir HERTER nie gelac.

HERTE *‹subst.›*
  HERTE
    MHT, 2, 18 und hât vil grôze HERTE an mir dâ her begangen

HERZE /HERZ s.a. -liep herzentrût -sêr -swaere
  HERZ
    PFE, 1, 38 dâ von mîn HERZ wart wunt.
    HOT, 5, 11 diu mîn niht gedenket*: daz krenket* HERZ und sinne mir.
  HERZE
    FEN, 2, 10 daz ich daz HERZE von ir iemer kêre.
    FEN, 3, 13 wan diu mir kundez HERZE alsô versêren,

HERZELIEP *‹subst.›*
  HERZELIEBE
    WVK, 5, 11 swer bî HERZELIEBE tougen lît.
  HERZELIEP
    UVS, 10, 14 hôhen muot noch HERZELIEP niht vinden kan,
    WIN, 7, 18 und er niht wan HERZELIEP gewinnet.

HERZELIEP *‹adj.›*
  HERZELIEBE
    HVF, 3, 18 sît ez diu HERZELIEBE wil
  HERZELIEBIU
    UVS, 3, 9 HERZELIEBIU vrouwe+,
  HERZEN LIEBEN
    MHT, 1, 10 von eines HERZEN LIEBEN wîbes güete enpfangen,

HERZESÊR /-SÊRE
  HERZEN SÊRE
    KSL, 1, 46 diu senden HERZEN SÊRE
  HERZESÊR
    UVS, 19, 16 in iemer wernde HERZESÊR.
    DVG, 3, 8 gewinnen iemer HERZESÊR
  HERZESÊRE
    UVS, 1, 17 al mîn HERZESÊRE,
    UVS, 14, 25 dâ wart ein lieplîch kus nâch HERZESÊRE,

Fig. 2

De volgende stap die we experimenteel willen wagen is, deze concordantie als Arbeitswörterbuch voor verdere "automatische" lemmatisering van lyrische teksten uit de 13e eeuw te gebruiken. Dat we ondanks het Bonner échec hier over de mogelijkheden wat optimistisch zijn, komt door het topische karakter van de 13e eeuwse Duitse lyriek. Deze lyriek lijkt zoveel op elkaar dat een hoger succes-percentage dan in Bonn te verwachten is.

O. Janssen
A.H. Touber

Bijdrage van de computer-afdeling:
- produceren van een volledige index
- ontwikkelen van een methode om interaktief (aan een terminal zittend en door middel van invulling in het tekst-fragment op het beeldscherm) individuele voorkomens-aanduidingen te voorzien van een verwijzing naar een andere voorkomens-aanduiding, die hierna als lemma wordt beschouwd
- produceren van een nieuwe versie van de index, waarin alle voorkomens-aanduidingen zijn gerangschikt onder de juiste lemmata
- afdrukken van de volledige concordantie uitgaande van de oorspronkelijke tekst en de laatste versie van de index waarin (weer_interaktief) volgens een bepaalde afspraak-verwijzingen, woordsoort-aanduidingen en verklaringen zijn aangebracht.

SPRAAKSIGNAAL ANALYSE

Inleiding.

Bij spraaksignaal analyse binnen het Instituut voor Fonetische Wetenschappen IFA wordt de computer gebruikt om uit het spraaksignaal informatie te halen over de vorm van de mond-keelholte en over bepaalde karakteristieken van het signaal zelf, zoals de formanten. In het hierna volgende worden vier belangrijke onderdelen van de spraaksignaal analyse aangestipt:
- de modellering van de spraakbuis
- het digitaliseren van een analoog spraaksignaal
- de lineaire prediktie methode (LPC)
- de Fast Fourier Transform (FFT).

Modellering van de spraakbuis.

De vorm van de mond-keelholte is bepalend voor de aard van het spraaksignaal. Als we inzicht willen krijgen in de relatie tussen die twee, dan moeten we er voor zorgen dat de spraakbuis zo is gemodelleerd dat de wiskundige formulering van de spraakproduktie niet nodeloos ingewikkeld wordt.
Een abstraktie van de spraakbuis die aan dat kriterium voldoet is het zg. n-buis model: een aaneenschakeling van n cylinders van gelijke lengte (zie fig. 3).

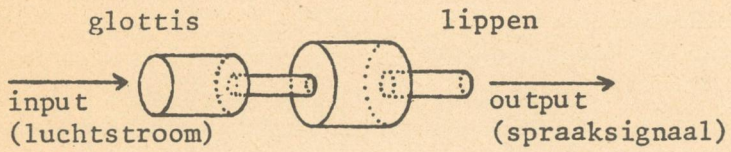


Fig. 3
een n-buis model met 4 segmenten

Met behulp van een wiskundig model dat de voortplanting van geluidsgolven door zo'n n-buis beschrijft kunnen we op numerieke wijze de invloed van artikulatoren zoals de tong en de lippen op de aard van het spraaksignaal bestuderen.

Het uitdunnen van het aantal gegevens.

Een spraaksignaal kunnen we opvatten als een kontinue, oneindige stroom van gegevens. Als we met de computer spraakanalyse willen verrichten, dan zullen we er dus voor moeten zorgen dat die toevloed eindige proporties heeft zodat hij numeriek hanteerbaar is. Een veelgebruikte methode om de hoeveelheid gegevens in te perken is het zg. bemonsteren van het signaal ("sampling"). Het analoge signaal op de band wordt daarbij periodiek, d.w.z. om de T sekonden, op z'n waarde onderzocht. De zo gevonden rij van waarden (het digitale signaal) is eindig (zie fig. 4).

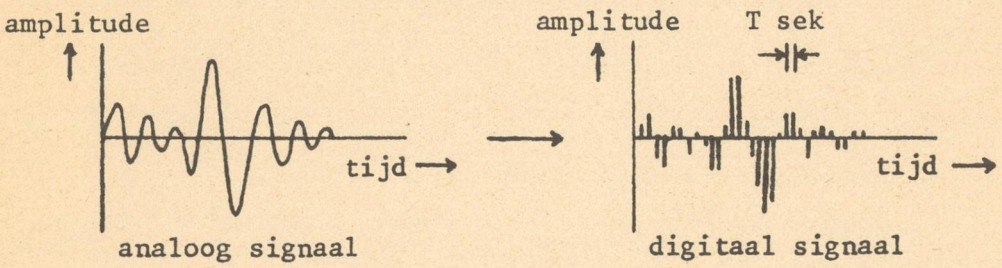


Fig. 4
het bemonsteren van een signaal

De sample-frekwentie, het aantal samples dat er per sekonde uit het signaal wordt gelicht, mag niet te laag zijn omdat er dan essentiële informatie over het signaal verloren gaat. De omzetting van analoog signaal in digitaal signaal wordt aangeduid met de term A/D-konversie.

De rol van LPC in de spraaksignaal analyse.

Een belangrijke methode bij het analyseren van een spraaksignaal is de lineaire prediktie methode. De diverse modifikaties van deze methode worden samenvattend aangeduid als LPC, Linear Predictive Coding of Speech.
De lineaire prediktie methode is afkomstig uit de cybernetika, de wetenschap die zich bezighoudt met de mathematische modellering van terugkoppelings mechanismen in onder andere fysische en biologische processen. Bij deze methode wordt verondersteld dat het signaal de eigenschap heeft dat de waarde op een bepaald tijdstip lineair afhankelijk is van een aantal voorafgaande signaal-waarden, met andere woorden dat er een kombinatie van getallen a1, ..., aM bestaat zo dat

$$\langle s(n) \rangle = a1\ s(n-1) + a2\ s(n-2) + \ldots + aM\ s(n-M)$$

een goede benadering vormt van de (gemeten) signaal-waarde s(n) (zie fig. 5).

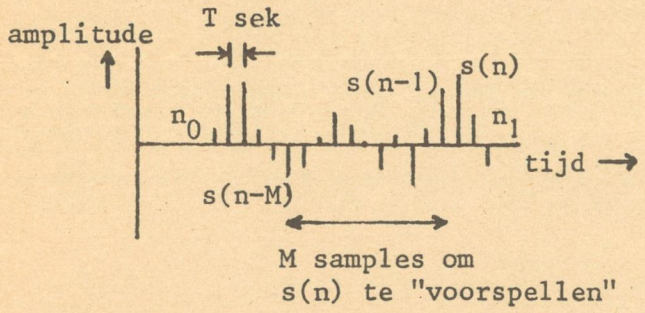


Fig. 5
M-de orde lineaire prediktie van het sample s(n)

Als we een bepaald interval, bijvoorbeeld van n0 tot n1, willen analyseren, dan moeten we er voor zorgen dat in dat interval ieder sample zo goed mogelijk lineair afhankelijk is van zijn M voorgangers. Met behulp van optimaliserings methoden kunnen we de meest geschikte koëfficiënten a1, ..., aM vinden.
Door deze optimalisering is het aantal gegevens verder verkleind tot de M koëfficiënten a1, ..., aM. Ze vormen het "recept" waarmee het signaal kan worden opgebouwd. De getallen a1, ..., aM karakteriseren de wijze waarop de mond-keelholte een bepaald spraakgeluid tot stand brengt.
Enerzijds kunnen we met deze zg. filterkoëfficiënten het doorsnedeverloop van de mond-keelholte in termen van een n-buis berekenen, en anderzijds de formanten en bijbehorende bandbreedten van het betreffende signaal-segment.

Bepaling van het spektrum van een signaal.

Het spektrum van een signaal is de verschijningsvorm die ontstaat na een bepaalde wiskundige transformatie van het signaal, de zg. Fourier-transformatie. Deze Fourier-transformatie weegt in het signaal hoe sterk elk van de optredende frekwenties meedoet. Het spektrum laat dus zien hoe zwaar elk van de sinus-vormige trillingen in het signaal is vertegenwoordigd. De formanten kunnen we vereenzelvigen met de pieken in het spektrum. De berekening van het spektrum geschiedt met behulp van een aan de computer aangepaste versie van de Fourier-transformatie, de Fast Fourier Transform (FFT).
Het is mogelijk een goede schatting van het signaal-spektrum te geven door de FFT toe te passen op de genoemde filterkoëfficiënten a1, ..., aM. Dit spektrum wordt LPC-spektrum genoemd.
Het spektrum kan worden gebruikt om de formanten te berekenen. We moeten in dat geval op numerieke wijze de frekwenties bepalen waarvoor er in het spektrum een piek optreedt.

Het spraaksignaal analyse systeem.

De voor spraakanalyse in het IFA aanwezige programmatuur is een onderdeel van de zg. ILS-programma bibliotheek. Deze bibliotheek bevat velerlei programmatuur op het gebied van signaal manipulatie.
De struktuur van het spraaksignaal analyse systeem is weergegeven in fig. 6.
Bij de voorbewerking van het signaal wordt onder andere gekorrigeerd voor de effekten die kunnen optreden ten gevolge van het afkappen van een signaal.

Voorbeelden.

Het eerste voorbeeld geeft de resultaten van analyse van een gedeelte van een signaal dat is verkregen door aanblazing van een perspex n-buis model met de zg. klinkersirene.
De analyse is gestart bij het in fig. 7 met A aangegeven interval. Het analyse-interval schuift telkens 1 schaaldeel op naar rechts. In fig. 8 zijn voor 10 achtereenvolgende analyse-intervallen de m.b.v. LPC gereconstrueerde buizen getekend. De buizen zijn zo

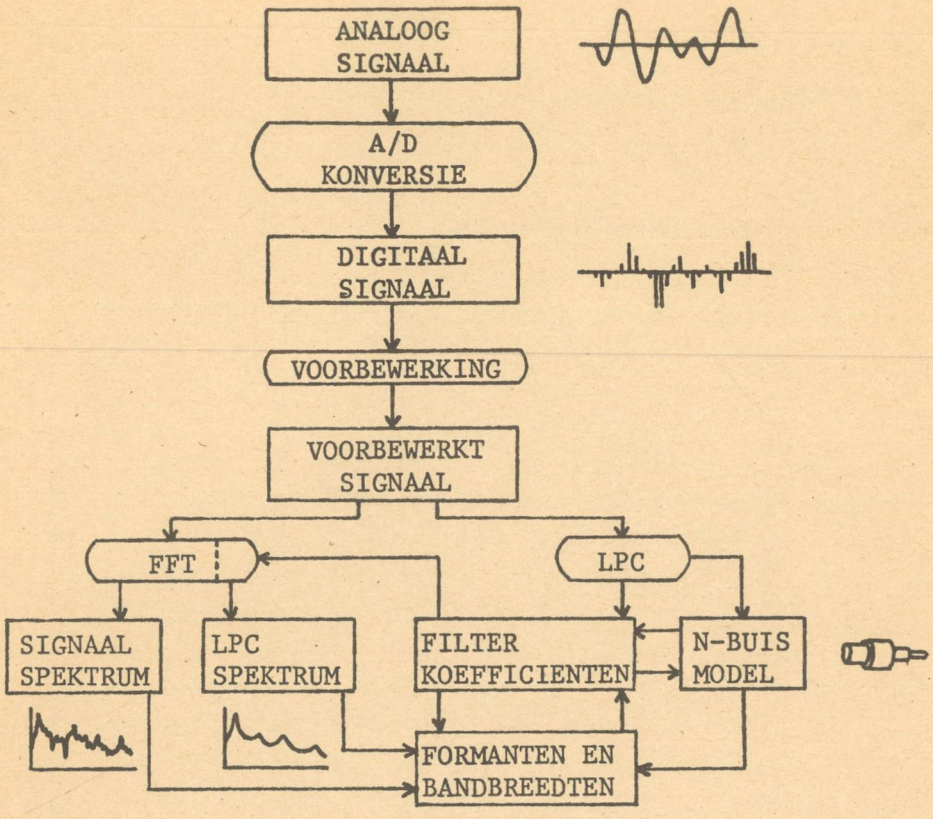


Fig. 6
het spraaksignaal-analysesysteem

geschaald dat de segmenten aan de linker uiteinden gelijke doorsnede hebben. De lengte van de perspex buis is in werkelijkheid 19,1 cm. In fig. 9 zien we de LPC-spektra van de zelfde opeenvolgende analyse-intervallen. Fig. 10 laat het verloop van de formanten en bandbreedten (de vertikale strepen) zien als functie van de tijd. Dit diagram is een soort bovenaanzicht van achter elkaar geplaatste spektra. Tenslotte is in fig. 11 het signaal-spektrum te zien met het voorbewerkte signaal waaruit het afkomstig is. De overeenkomst tussen de LPC-spektra en dit spektrum is duidelijk te zien.
Het tweede voorbeeld is de rekonstruktie van een 4-buis als formanten en bandbreedten gegeven zijn. In fig. 12 is te zien welke 4-buis met een lengte van 17,5 cm hoort bij een bepaald formantenpaar F1, F2 met gegeven bandbreedten (10 Hz). De getekende buizen kunnen worden beschouwd als (eenvoudige) modellen van de spraakbuis voor de produktie van klanken in de klinkerdriehoek.

L.J. Bonder

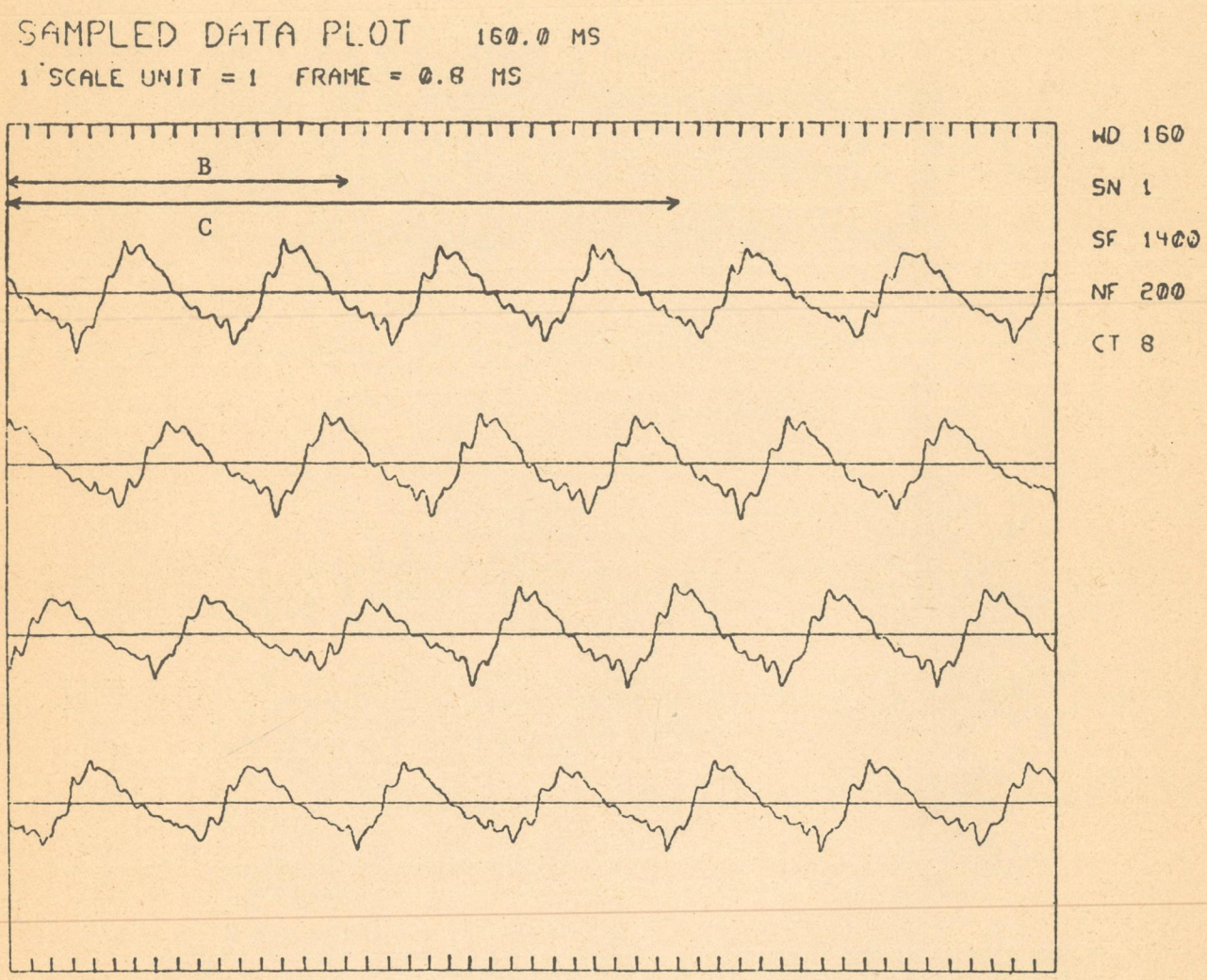
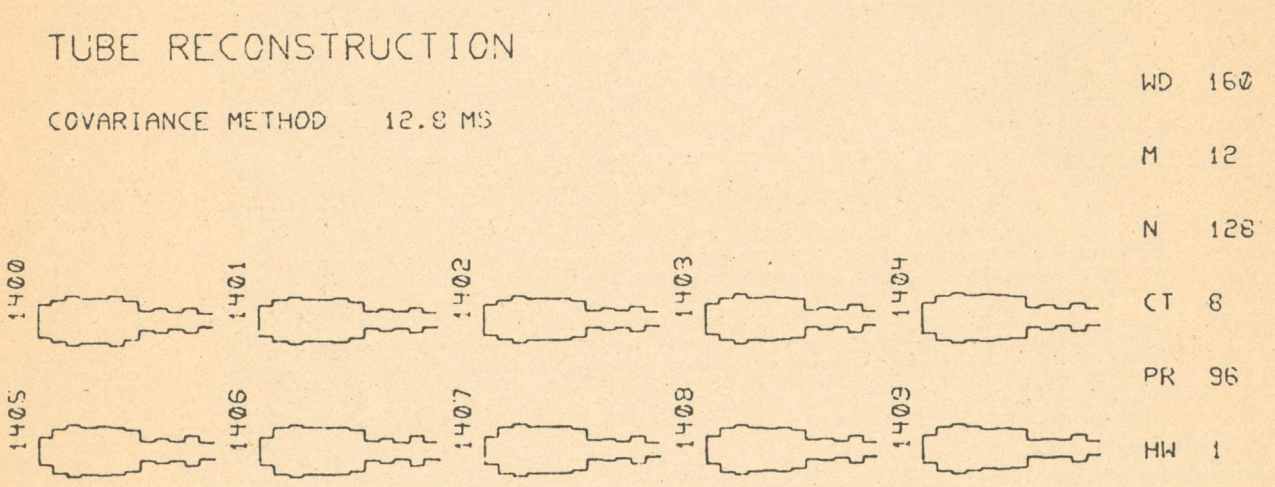


Fig. 7



Ter vergelijking de afmetingen van de oorspronkelijke hardware buis:

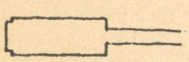


Fig. 8

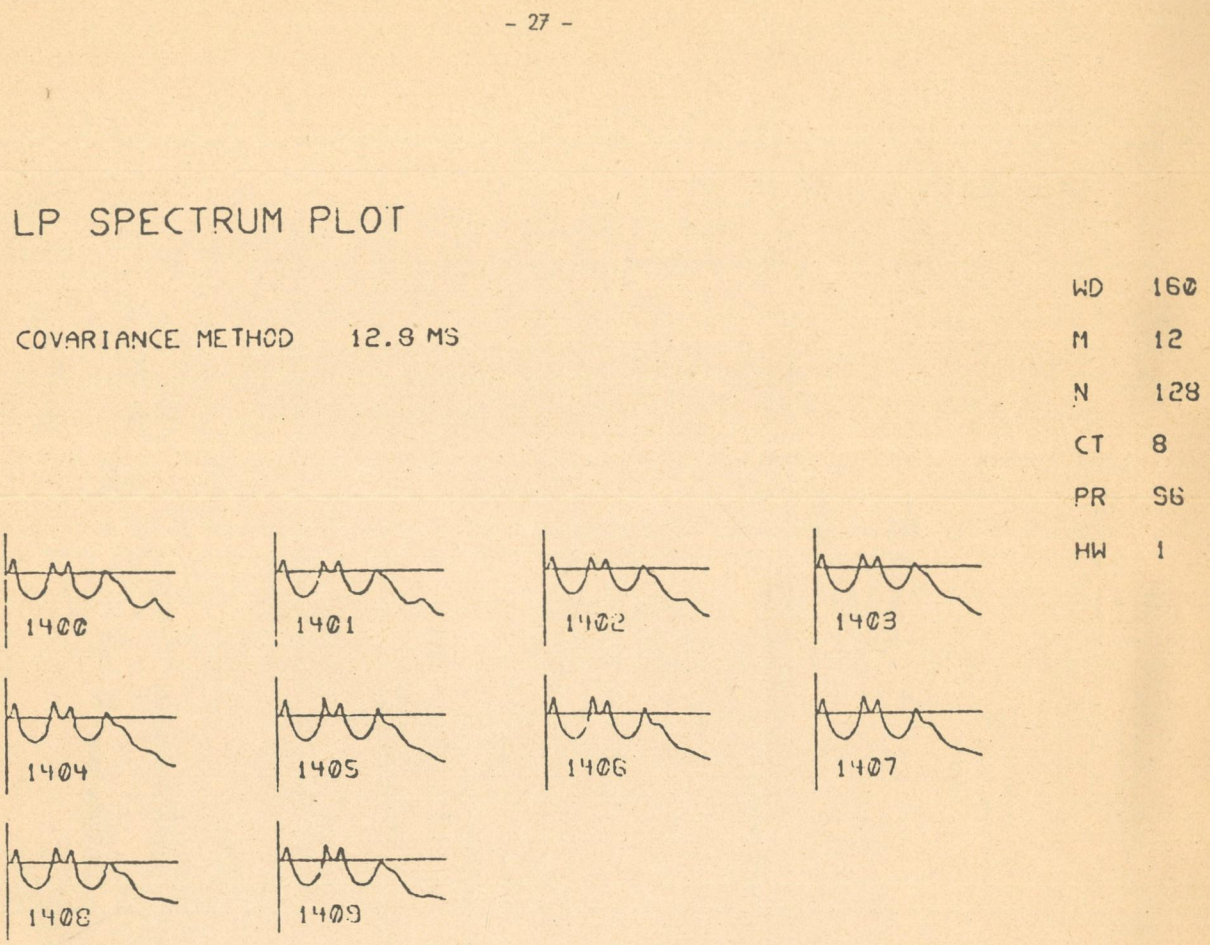

LP SPECTRUM PLOT
COVARIANCE METHOD 12.8 MS
WD 160
M 12
N 128
CT 8
PR 96
HW 1
1400
1401
1402
1403
1404
1405
1406
1407
1408
1409

Fig. 9

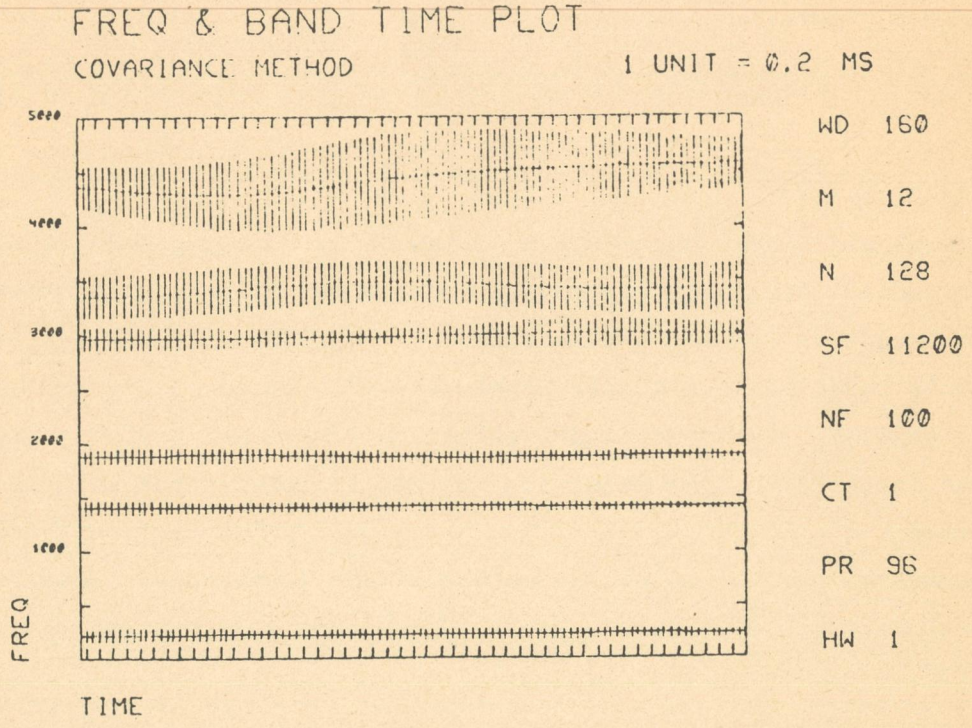

FREQ & BAND TIME PLOT
COVARIANCE METHOD
1 UNIT = 0.2 MS
WD 160
M 12
N 128
SF 11200
NF 100
CT 1
PR 96
HW 1
FREQ
TIME

Fig. 10

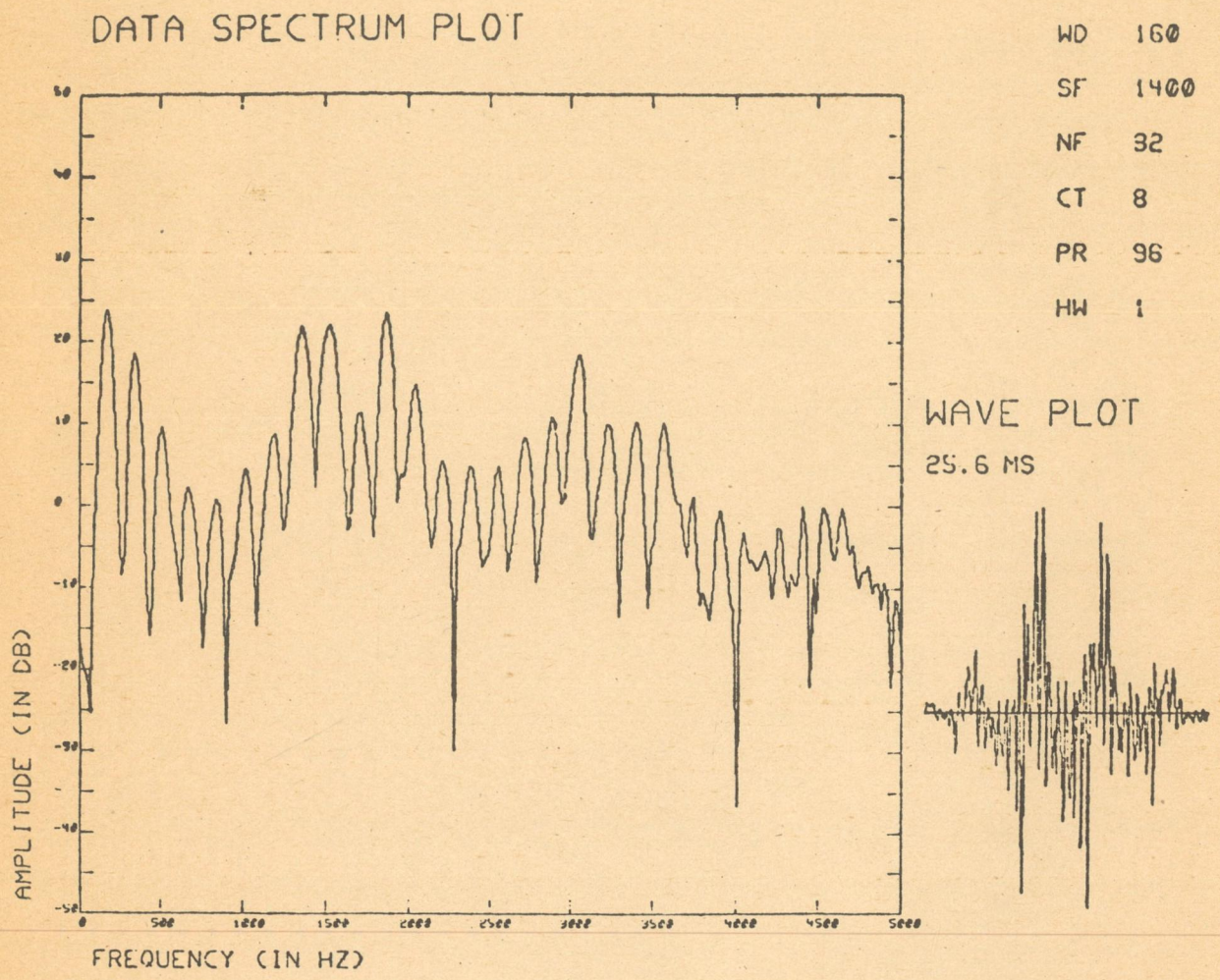


Fig. 11

Voorbeeld 2: fig.12.

In dit voorbeeld worden de n-buis modellen berekend uit gegeven formanten en bandbreedten. Uitgaande van twee formanten met bijbehorende bandbreedten kunnen we dan een 4-buis model rekonstrueren. In fig.12 zien we welk 4-buis model met een totale lengte van 17,5 cm hoort bij een bepaald formantenpaar (F_1 ,F_2) met bandbreedten van 10 Hz. Ook hier zijn de buizen zo geschaald dat de segmenten aan het linkeruiteinde gelijke doorsnede hebben.

De getekende buizen kunnen worden beschouwd als (eenvoudige) modellen van de spraakbuis voor de produktie van klanken in de klinkerdriehoek (vgl. Mol, 1973).

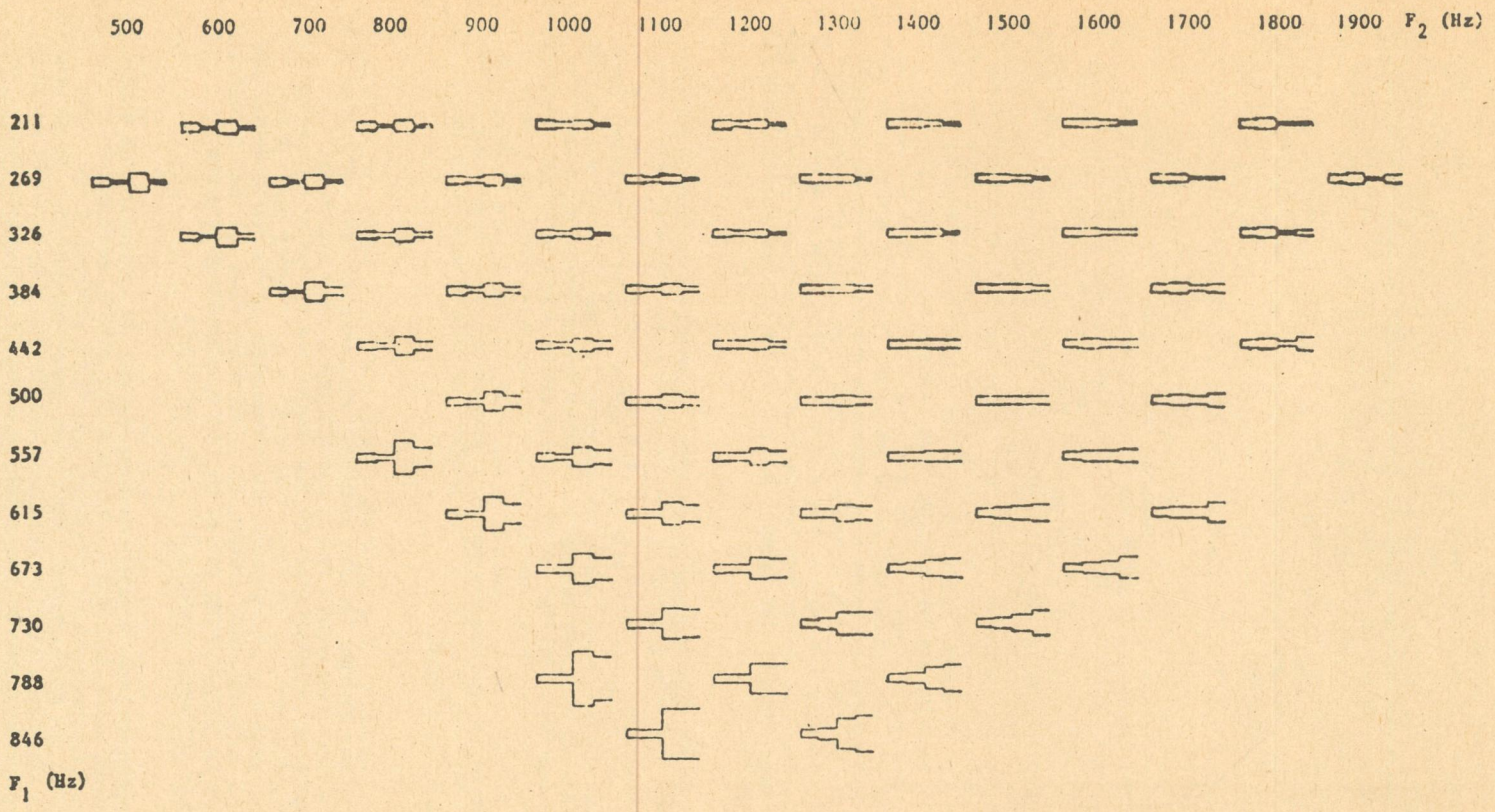

500
600
700
800
900
1000
1100
1200
1300
1400
1500
1600
1700
1800
1900
F2 (Hz)
211
269
326
384
442
500
557
615
673
730
788
846
F1 (Hz)

Fig. 12

De notering van maatloze muziek.

In de muziekwetenschap speelt het muziekschrift een belangrijke rol. Het is een bruikbaar middel om muzikale composities te bestuderen, want het stelt musicologen in staat zich een klankvoorstelling te maken van een muziekstuk zonder dat het betreffende stuk daadwerkelijk wordt gespeeld. Daar komt nog bij dat zeer veel notaties door de componist zelf zijn gemaakt; zij geven daardoor een beeld dat helder aantoont wat de componist wilde / kon noteren, en dat niet vertroebeld is door de interpretatie van een uitvoerder. Tenslotte is het notenschrift een handzaam, zij het niet volkomen hulpmiddel om van ongenoteerde muziek een bruikbare beschrijving te maken. Veel volksliederen zijn alleen middels notaties achteraf bewaard gebleven en kunnen nog slechts met behulp van deze notaties worden bestudeerd. Het notenschrift heeft echter ook zijn beperkingen. De grootste is wel dat alleen muziek met een duidelijke maat goed noteerbaar is. Voor zogenaamde maatloze muziek is het muziekschrift ontoereikend. Dit houdt in dat alle muziek die niet gebaseerd is op een of ander maatpatroon, op grond van een notatie, niet of nauwelijks wat haar temporele eigenschappen betreft bestudeerbaar is.

Enige jaren geleden werd ik voor het eerst met dit probleem geconfronteerd toen ik de tijdstruktuur van de zg taqsim wilde bestuderen. Onder taqsim wordt in de Arabische muziek een wijze van improviseren verstaan, waarbij wel in zekere zin een bepaald voorgeschreven toonhoogten-verloop wordt aangehouden, maar waarbij de tijdsordening volledig vrij wordt gelaten. Om de tijdsindeling van dergelijke stukken, waarvan ik er inmiddels vele had verzameld, te kunnen bestuderen, zocht ik naar een middel om van de opeenvolgende tonen de hoogte en de duur te meten. Niet gehinderd door een te veel aan kennis op het terrein van de natuurkunde of de techniek ben ik bij diverse instituten op zoek gegaan naar het apparaat dat een dergelijke registratie even zou volbrengen. Tot mijn verbazing en teleurstelling bleek een dergelijk apparaat niet te bestaan en tot overmaat van ramp werd ik door behulpzame collegae er steed weer op gewezen dat het probleem waar ik mee aankwam zeer gecompliceerd was. Het heeft enige tijd en ik denk ook wel moeite gekost voordat men mij vooral van dit laatste had overtuigd. Toen begon het geduldige en moeizame proces van onderzoek of en zoja, hoe het mogelijk is om van een eenvoudig, op tape opgenomen, melodietje de lengte en hoogte van de opeenvolgende tonen te meten. Ik zal dat proces maar niet beschrijven en volstaan met de opmerking dat tenslotte een systeem is ontwikkeld dat voldoet aan de gestelde eisen. Het systeem is ontwikkeld voor de huidige computer-configuratie van de FdL maar kan zonder veel problemen aangepast worden aan andere configuraties. De werking van het systeem zal ik hieronder uitleggen door de opeenvolgende bewerkingsfases te beschrijven die een melodietje ondergaat van het moment dat het op een tape is opgenomen tot het ogenblik dat het er als een stukje muziekschrift uitrolt.

Fase 1.
Een stukje muziek dat op een tape is opgenomen wordt afgespeeld op een aan de computer aangesloten taperecorder. Een klein apparaatje, dat analoog-digitaal-converter (ADC) heet, zorgt ervoor dat het signaal dat van de tape wordt gelezen, 'vertaald' wordt in een reeks getalletjes. Dit vertalen kunnen we ons gemakkelijk voorstellen wanneer we bedenken dat geluid geregistreerd kan worden als een kurve (vgl de groef van een grammofoonplaat). Van zo'n kurve kunnen we dan op elk gewenst tijdstip aangeven hoever die uitwijkt t.o.v. een aangenomen nullijn. Op deze wijze kunnen we het verloop van de kurve in getallen weergeven. We moeten erg veel metingen verrichten om het verloop van de kurve nauwkeurig weer te geven. De ADC is een handig apparaatje dat dergelijke metingen vlug en betrouwbaar kan verrichten. Voor een muzieksignaal met een maximale bandbreedte van 20.000Hz zijn 40.000 metingen per seconde nodig. Het door de ADC in getallen vertaalde signaal wordt dan in het geheugen van de computer opgeslagen.

Fase 2.
In deze fase wordt een zogenaamde Fourier-transformatie uitgevoerd. Dit is een bewerking die berust op de stelling van Fourier dat elke periodieke golfbeweging, hoe ingewikkeld deze van vorm ook is, kan worden voorgesteld als een opeenstapeling van enkelvoudige harmonische golven. Omdat een muzikale klank tot het domein van het geluid behoort en omdat geluidsgolven periodiek zijn, is het volgens de stelling van Fourier mogelijk een muzikale klank uiteen te rafelen in zijn samenstellende deeltonen. De Fourier-transformatie nu is een procedure, die de grote reeks getallen die in de vorige fase zijn vervaardigd van het geluidsignaal, in gelijke blokken knipt en van elk blok getallen berekent welke verschillende enkelvoudige trillingen (toonhoogten) er in voorkomen.

Fase 3.
Een muzikale toon is opgebouwd uit verschillende komponenten, de zg natuurtonen. Deze zijn gemakkelijk waar te nemen in de klank van een kerkklok. Bij deze natuurtonen wordt gewoonlijk een onderscheid gemaakt in de grondtoon, die vooral bepalend is voor onze toonhoogte-gewaarwording, en de boventonen, die eerder timbre-bepalend zijn. In de praktijk blijkt echter dat in een muzikaal verloop de grondtoon van een klank fysisch niet aanwezig hoeft te zijn om toch gehoord te worden. Ons gehoor is kennelijk in staat om deze ontbrekende component van een klank aan te vullen of te reconstrueren. Een dergelijke reconstructie wordt in deze fase uitgevoerd. Deze komt erop neer dat van de opeenvolgende geanalyseerde klanken wordt nagegaan of steeds de onderste komponenten (de grondtonen)

wel een 'melodische' verloop vormen zonder grote of 'onlogische' sprongen. Op de punten waar dergelijke afwijkingen voorkomen, wordt gezocht naar een alternatieve grondtoon die past in het horizontale verloop van de opeenvolgende grondtonen, en die op grond van de boventonen van de betreffende klank mogelijk is. Uit dit programma van wikken en wegen komt dan de globale muzikale lijn van het te analyseren stukje muziek naar voren.

Fase 4.
Het moeilijkste werk is nu gedaan. Wat nu nog moet gebeuren is het samenvatten van de opeenvolgende grondtonen van (bijna) gelijke frekwentie tot grotere eenheden: de tonen. Deze verschijnen in de uiteindelijke uitvoer gespecificeerd naar toonnaam (A, B, C, etc), naar oktaaf en naar toonduur. Bovendien wordt per toon nog een aantal gegevens vermeld waaruit op te maken valt hoe geloofwaardig de opgegeven waarden zijn. Omdat musicologen er een hekel aan hebben om muziek als een tabel te moeten lezen, is aan deze fase nog een schakel toegevoegd waardoor het mogelijk is om de gegevens in muziekschrift uit te voeren. Het systeem is eerst onlangs gereed gekomen en het is nog niet volledig uitgetest. Wanneer dit gebeurd is, zal het systeem in eerste instantie worden gebruikt voor het onderzoek naar de temporele eigenschappen van de taqsim, dat zijn arabische maatloze solo-improvisaties. De uitgangs-hypothese daarbij is dat ook maatloze muziek beschreven kan worden in termen van regelmatige pulsen, met dien verstande dat de vrijheidsgraad wat betreft de lengte tussen de opeenvolgende pulsen groter is dan bij 'gemeten' muziek en dat er vaker sprake is van maatwisseling.

Voor de totstandkoming van het systeem is gebruik gemaakt van bepaalde programmatuur die is ontwikkeld door de vakgroep Fonetiek en van programma's die ter beschikking zijn gesteld door Martin Piszczalski en Bernard Galler, beiden van de Universiteit van Michigan. De totstandkoming van het systeem is verder te danken aan Leigh Landy en vooral aan Leo Hennen, zonder wiens kennis op het terrein van de natuurkunde en de informatica ik nu nog zat te tobben.

Gebruikers van het systeem zijn welkom.

L.J. Plenckers
december 1980

A68-17S, een vrijwel kompleet komputerprogramma in Algol68 voor een deelgrammatika van een fragment Engels.

Van 15 tot en met 17 maart 1979 werd te Milwaukee, Wisconsin, VS, een konferentie gehouden getiteld On Current Approaches to Syntax. Een van de veertien bijdragen was afkomstig van S. C. Dik, die het vooraf gegeven korpus van 17 zinnen beschrijft volgens het door hem ontwikkelde model van Funktionele Grammatika.

De regels van deze grammatika zijn redelijk formeel en expliciet, zoals dat de laatste decennia de gewoonte is in de theoretische linguistiek. Bij het schrijven van een grammatika is een linguist echter net zo onderworpen aan beperkingen als een menselijke taalgebruiker bij het produceren van taaluitingen: ook hier bestaat onderscheid tussen "performance" en "competence". Om de samenhang van enigszins komplexe grammatikale voorschriften en vooral de uiterste konsekwentie er van te kunnen nagaan, m.a.w. om te zien of de grammatika echt expliciet is en ook adekwaat, is het niet voldoende zich met pen en papier achter de schrijftafel te zetten. Om het uitgewerkte fragment van een Funktionele Grammatika voor het Engels - de Sample Grammar uit Dik's "Seventeen Sentences" - te testen, heb ik een komputerprogramma bedacht dat die grammatika zo getrouw mogelijk moet simuleren. Hieronder volgt in zo algemeen mogelijk gehouden bewoordingen een schets van gegevens-strukturen en algoritme van dat programma. Het is noodzakelijkerwijs een grove, sterk vereenvoudigde en onvolledige weergave van de werkwijze van de Funktionele Grammatika waarin volstrekt geen recht wordt gedaan aan de theorie en zijn achtergronden.

Het programma is geschreven in Algol68, o.m. om een eventuele publikatie-versie er van zo leesbaar mogelijk te laten zijn; het is gedraaid op de apparatuur van SARA (CDC Cyber 73-28 en 174-16). Aan dit projekt is, tussen half november 1978 en 31 augustus 1979, in totaal vier tot vijf maanden werk besteed. Daarna is een vervolgprojekt van start gegaan, dat in de zomer van 1980 heeft geleid tot een publikatie over de vorming van beperkende betrekkelijke bijzinnen waarin (subjekt- of objekt-) raising heeft plaatsgevonden op de gerelativiseerde term.

De voornaamste gegevens-strukturen zijn: predikaatschema, term en predikatie. Het lexikon is een opsomming van de predikaatschema's die een basis-predikaat bevatten.

De hoofdlijnen van de algoritme zijn:

1) Kies uit het beschikbare lexikon een verbaal predikaatschema (d.w.z. een werkwoord) met een of meer argumentplaatsen.
2) Zet al dan niet termen op de eventueel aanwezige plaatsen voor instrumentele bepaling en/of redengevende bijzin.
3) Zet termen op de overige argumentplaatsen, een voor een, in willekeurige volgorde, tot er precies een plaats open blijft.
4) Zet een term op de laatstovergebleven open argumentplaats (zie ook 10).
5) Wijs een van de termen aan als subjekt, en eventueel ook een als objekt.
6) Druk alle konstituenten van de zin af, nadat zijn bepaald: aspekt en tijd van het werkwoord, morfologie, en volgorde.

Aanvullende opmerkingen hierbij:

7) Soms moet een term de gedaante van een bijzin aannemen: zo'n bijzin wordt verkregen door de bovenstaande stappen 1-5 opnieuw te doorlopen.
8) Anders wordt een term verkregen door uit het lexikon een nominaal predikaatschema (een naamwoord, voornaamwoord of eigennaam) te kiezen en daar eventueel aan toe te voegen: een lidwoord en/of een beperkende betrekkelijke bijzin.
9) Zo'n beperkende betrekkelijke bijzin wordt verkregen door nogmaals de stappen 1-5 te doorlopen.
10) Onder bepaalde voorwaarden moet of mag op de laatste open argumentplaats worden gezet: een betrekkelijk voornaamwoord, een vraagterm, of een anaforische term.
11) Er is een bepaald soort werkwoord dat steeds een ingebedde zin op zekere argumentplaats vereist; die term kan optreden als subjekt of als objekt; het is echter ook mogelijk deze syntaktische funktie toe te kennen niet aan de totale bijzin maar alleen aan het subjekt van die ingebedde bijzin: in zo'n geval neemt de bijzin de vorm aan van een infinitief-

konstruktie (transformationalisten noemen zoiets "raising").
Nevenschikking kan op diverse manieren worden gerealiseerd:
12) Konjunktie van twee of meer rijtjes argumenten bij 1 werkwoord (transformationalisten noemen zoiets "gapping"); bijvoorbeeld: "Simon schrijft een grammatika, en Tjoe Liong, een programma".
13) Konjunktie van termen, mits niet in de vorm van een bijzin.
14) Konjunktie van voliedige zinnen (van elke soort: hoofdzinnen, redengevende, ingebedde en betrekkelijke bijzinnen).

Relevante literatuur (verkrijgbaar bij het Instituut voor ATW):
S.C. Dik, Seventeen Sentences. Publikaties Inst.ATW nr 22, feb 1979.
Kwee Tjoe Liong, A68-FG(3). Publikaties Inst.ATW nr 23, maart 1979.
Met supplement A68-FG(7): versie 7, maart 1980.

Kwee T.J.
(Instituut voor Algemene Taalwetenschap)

Hoofdstuk 5. Welke ervaring en standaard-programmatuur hebben wij nu ?

In dit hoofdstuk belichten we aan de hand van de beschreven projekten in hoofdstuk 4 een taak van de computer-afdeling die voor gebruikers het meest interessant is : de ontwikkeling van standaard programma's. Zoals in hoofdstuk 1 al werd opgemerkt: het kan soms jaren duren voordat duidelijk is wat het gemeenschappelijke is in de vraagstelling van verschillende onderzoekers. Omdat onderzoeks-projekten aan bepaalde termijnen zijn gebonden moeten echter wel konkrete resultaten worden geproduceerd. Op deze manier ontstaat een groeiend aantal deel-programma's die, in eerste instantie ad-hoc geschreven, uitgroeien tot bouwstenen waarmee ook in andere situaties gewerkt kan worden. Een voortdurend groeiend deel-programma wordt echter onoverzichtelijk zodat zich mettertijd de noodzaak voordoet om een nieuwe, doorzichtige versie te schrijven. Deze laboratorium-situatie vraagt meer programmeer-tijd dan een gebruiker zich meestal realiseert.
Het is daarom belangrijk om, min of meer los van konkrete projekten, zich te blijven bezinnen op de inherente kenmerken van klassen van problemen wanneer zij met de computer moeten worden opgelost. Tot nu toe heeft het eigen onderzoek van de twee medewerkers zich daarop gericht.
In de vorige hoofdstukken is al naar voren gekomen dat veel onderzoekers in de faculteit te maken hebben met grote hoeveelheden gegevens die in vaak ongestruktureerde vorm aanwezig zijn in teksten, documenten, muziekstukken of gesproken taal. (Hierna voor het gemak aangeduid met "teksten": het gemeenschappelijke is namelijk dat het allemaal rijen van tekens zijn). De opslag van deze teksten vereist een aanzienlijke zorg en de computer-afdeling prijst zich gelukkig dit beheer in eigen hand te kunnen houden. De invoer, interne computer-opslag in termen van data-strukturen, bewerking van de gegevens en de uitvoer vragen allen om efficiency vanwege de veelheid en de complexiteit daarvan. Het is vaak niet van te voren aan te geven welke gegevens uit een tekst in een onderzoek gebruikt zullen worden, terwijl dit veelal een voorwaarde is voor het gebruik van reeds bestaande programmatuur. Gebruikt men bijvoorbeeld een data-base pakket dan moeten van te voren de relaties tussen de verschillende gegevens vastgelegd worden. Deze tegenstrijdigheid is de oorzaak van het lage rendement van de veelal moeizaam bij elkaar gebrachte en in een computer ingevoerde teksten. Het is uitermate belangrijk om op een simpele en goedkope manier vanuit deze "ruwe teksten" specifieke bestanden te kunnen maken, met alleen die gegevens die voor een bepaald onderzoek van belang zijn, en die direkt gebruikt kunnen worden in standaard-programma's (bijvoorbeeld SPSS of SIR).
Een formele manier om de struktuur van een tekst te kunnen beschrijven is om gebruik te maken van grammatica's. Een grammatica geeft in het algemeen aan hoe een rij tekens - de "tekst"- opgedeeld kan worden in de stukjes die voor de opsteller van de grammatica van belang zijn. Deze manier van beschrijven is kort en bondig en stamt uit de informatica: veel programmeertalen worden er mee beschreven. Toegepast op teksten kan men ze "bestands-grammatica's" noemen. Hieronder (B) is beschreven hoe op de computer-afdeling een systeem tot ontwikkeling is gebracht dat gestuurd wordt door dergelijke grammatica's en de hierboven aangegeven simpelheid en efficiency bij het omvormen van bestanden realiseert (en daarbij tevens fouten signaleert). Het systeem is eveneens te gebruiken bij de ontleding van bijvoorbeeld morfologisch gecodeerde zinnen in een natuurlijke taal. Speciale aandacht werd besteed aan het zo eenvoudig mogelijk kunnen opschrijven van een (kontekst-vrije) grammatica: het systeem is namelijk ook voor inferentie geschikt. De gebruiker stelt dan een vermoedelijke grammatica op voor de struktuur van teksten die hij al kant en klaar van anderen heeft gekregen. Om zijn grammatica uit te proberen op de tekst ontdekt hij al gauw de tekortkomingen er van en stelt zijn grammatica bij. Dit is een proces dat het beste interaktief achter een terminal kan plaatsvinden. Een dergelijk gebruik van grammatica's komt bij het definiëren van programmeertalen niet voor.
Is van te voren duidelijk dat de gegevens die moeten worden ingevoerd alleen door een bepaald standaard-programma zullen worden behandeld (bijvoorbeeld SPSS) dan is een eenvoudiger aanpak mogelijk. Een door ons ontwikkeld invoer-validatie programma wordt in A beschreven.
Ook ten aanzien van de uitvoer zijn de nodige programma's ontwikkeld. Elke computer-leverancier levert tegenwoordig tekst-opmaak programma's en een gebruiker kan op de computer-afdeling instruktie krijgen in het gebruik daarvan. Maar het met redelijke snelheid kunnen uitvoeren van afwijkende lettertypen heeft onze speciale aandacht en zorg gehad. Zie C.
De manier van opslag van grote hoeveelheden gegevens op het schijf-geheugen van een computer is van groot belang voor de efficiency van programma's die met deze gegevens werken. Wij beschrijven in D een efficiënt werkend "woordenboek"-systeem. Dit systeem liet zich ook gemakkelijk integreren met het hiervoor beschreven grammatica-systeem: bij het ontleden van zinnen in natuurlijke taal kan dan van een echt woordenboek gebruik gemaakt worden. Als illustratie van mogelijk gebruik nemen we aan het eind van dit hoofdstuk de onderzoeks-plannnen op van Meijs van de vakgroep Engels.
Ten slotte de verwerking van de gegevens die in teksten aanwezig zijn. Zoals eerder gezegd: de onderzoeker weet soms van te voren niet wat hij daarvan nodig heeft. Hij kan hooguit een ruwe indeling van zijn teksten aangeven (bijvoorbeeld bij tekst-corpora: paragrafen, zinnen, woorden). Om hieraan tegemoet te komen werd een patroon- herkennings-programma geschreven, (E). Ook hier was efficiency en gebruikers-gemak een belangrijk ontwerp-criterium.
We zijn er een beetje trots op dat alle beschreven programma's werken en in de praktijk beproefd zijn. Voor programmeurs op de faculteits-computer zijn zij als afzonderlijke

procedures beschikbaar. Meer gedetailleerde beschrijvingen zullen verschijnen in technische rapporten.

A. Een systeem voor VALidatie van INvoer.

Invoer voor een computer klaarmaken kan vervelend zijn, en is dat vooral als het gaat om het ponsen van grote aantallen ponskaarten. Vervelende zaken veroorzaken vergissingen die het geheel nog vervelender maken, omdat een kleine fout het noodzakelijk maakt de hele kaart opnieuw te ponsen. Vergissingen treden natuurlijk het meest op bij het ponsen van hele reeksen cijfers, op onregelmatige plaatsen eventueel onderbroken door een willekeurig aantal spaties (blanco ponsingen). Dit soort voor fouten zeer gevoelige kaarten zijn juist nodig bij het prepareren van een "job" met het statistische programma-pakket SPSS op SARA. Voor de meeste gebruikers moeten deze in grote aantallen worden vervaardigd.
De computer-afdeling heeft een systeem ontwikkeld waarbij de gebruiker niet aan de ponsmachine maar aan de terminal van de faculteits-computer zijn gegevens kan intypen, waarbij een programma erop toeziet dat de gegevens die ingetypt worden binnen door de gebruiker gestelde specifikaties vallen. Hierbij wordt wel niet een uit echte kartonnen van gaten voorziene kaarten samengesteld bestand opgebouwd, doch wel een eenduidige afbeelding daarvan op de "schijf" van de Eclipse-computer, die voor alle praktische toepassingen bruikbaar is alsof het om een stapel kaarten ging en die als voordeel heeft, dat korrekties kunnen worden toegepast zonder dat meteen de hele kaart moet worden overgeponst.
De door de gebruiker gestelde specifikaties zijn van de vorm: "Voor het gebied van de kaart ("veld") tussen positie ("kolom") 4 en positie 8 mogen alleen gebruikt worden karakters uit de rij: 0123456789 (dus alleen cijfers)." Of: "het gebied tussen kolom 12 en kolom 16 moet voor iedere kaart blanco blijven."
Om het de gebruiker gemakkelijk te maken, kan hij elk veld een naam geven ("omschrijving van het veld"), terwijl zo'n veld mag zijn onderverdeeld in subvelden, elk met een eigen verzameling toegestane karakters.
Het opgeven van deze specifikaties gebeurt in een sessie met het programma VALID, waarbij medewerkers van de computer-afdeling behulpzaam zijn.
Het werkelijk aanmaken van het "data-bestand" wordt gedaan met het programma VALIN. Dit biedt de mogelijkheid tijdens een sessie verschillende kaartsoorten te produceren (dus met behulp van verschillende specifikaties). De gebruiker krijgt steeds een door hem zelf opgegeven naam voor een veld te zien, gevolgd door een rijtje onderstrepingen waarop hij de karakters die op dat veld moeten komen kan intypen. Zodra hij een fout heeft gemaakt, dus buiten de specifikaties komt, wordt opnieuw invoer voor hetzelfde veld gevraagd. Nadat alle velden van de "kaart" zijn ingevuld, wordt het hele kaartbeeld getoond en gevraagd of dit zo naar de zin van de gebruiker is. Na bevestigend antwoord kan de volgende kaart gemaakt worden. Dit systeem is door verscheidene gebruikers al met succes gebruikt en alhoewel het wat langzamer gaat dan bij een geoefende ponstypist(e), is het een stuk veiliger voor wat de correctheid betreft.

B. Parser-generator-systeem.

Zoals het programma VALIN / VALID gezien kan worden als een "vermenselijking" van het invoeren van gegevens voor programma-pakketten, zo kan het programma "LALRPAT" getypeerd worden als een "computerisering" van menselijke taal. Het gaat daarbij om het in de computer brengen van gegevens, afkomstig uit teksten, waarbij de relaties van te voren niet vastliggen maar pas in de loop van het onderzoek gespecificeerd zullen worden.
Voorbeelden: het opbouwen van historische gegevens-bestanden, zoals in het onderzoek van mevr. Wijzenbeek (zie hoofdstuk 4) of het vormen van een tekst-corpus met aantekeningen van de onderzoeker, zoals in de Hettitische teksten van Houwink ten Cate, of, in eenvoudiger vorm, in het Eindhovense Corpus. De "computerisering" betreft dan het ontwerpen van een formele taal die zo dicht mogelijk aansluit bij de oorspronkelijke tekst-indeling. Op de computer-afdeling wordt daartoe gebruik gemaakt van een ontleder voortbrenger (of "parser-generator" in goed Engels) voor kontekst-vrije talen. Deze test of de grammatica geen ambiguiteiten bevat en brengt een programma voort dat de aangeboden invoer in nette stukjes hakt die dan in het algemeen in een boom-struktuur worden opgeborgen. Op deze struktuur kunnen dan huidige of toekomstige programma's werken. De programma-naam LALR is een aanduiding voor de gevolgde ontledings-methode. Hierbij is in het begin dankbaar gebruik gemaakt van een gelijknamige parser-generator, geschreven in BCPL, van Jan van Katwijk, vakgroep Informatica, TH-Delft. Hiermee kunnen de meeste niet-ambigue kontekst-vrije grammatica's worden behandeld (ook links- en rechts-recursieve).
Om tegemoet te komen aan de wensen van een werkgroep van Algemene Taalwetenschap werd een uitbreiding gekreëerd voor zogenaamde patroon-grammatica's: de grammatica beschrijft dan een deel van de invoer (die in dit geval een nederlandse zin is).
In dit type grammatica treden echter vaak ambiguïteiten op. Een verdere uitbreiding van het programma kan ook deze situatie aan (geen backtracking maar splitsing van de onderliggende automaat). Hiermee is tevens de kunstmatige ambiguiteit opgevangen die ontstaat door het gat tussen C.F.-grammatica's en LALR grammatica's.
In de grammatica mogen reguliere expressies worden gebruikt. Verder mogen "don't care" symbolen gebruikt worden. Voor wat betreft het lexicale niveau (het uitzoeken wat de "woorden" zijn in de tekst) wordt gebruik gemaakt van het onder D beschreven lexicon-systeem. Hierin kan incrementeel gezocht worden. Op lexicaal niveau kunnen reguliere expressies van verzamelingen van karakters opgegeven worden. Het keyword "string" betreft

een willekeurig lange string. De beëindiging daarvan wordt door de parser-generator bepaald op grond van de symbolen die op "string" kunnen volgen. Op het lexicale niveau blijft de parser deterministisch werken, hoewel hij zijn verwachting haalt uit alle automaten die op dat moment parallel werken.

C. Afdrukken van teksten in afwijkende lettertypen.

De computer-afdeling heeft een systeem ontwikkeld dat aan de gebruikers de mogelijkheid geeft om een eigen verzameling afdruk-tekens (karakter-verzameling) te definiëren en een tekst te laten afdrukken in die tekens.
Er wordt momenteel met succes gebruik gemaakt van het systeem waarvan hierbij enkele voorbeelden getoond worden. Een karakter-verzameling van "schrijfletters" levert redelijk goede "met de hand geschreven" tekstjes op. Een andere, veel uitgebreidere karakter-verzameling, bestaande uit zes verschillend geschreven alfabetten en drie verschillende reeksen van cijfers alsmede leestekens en speciale tekens, is in gebruik bij het projekt Hettitisch van Houwink Ten Cate. (Zie hoofdstuk 4). Als voorbeeld laten we hierbij in figuur 13 een fragment Roemeens in Cyrillische tekens zien.

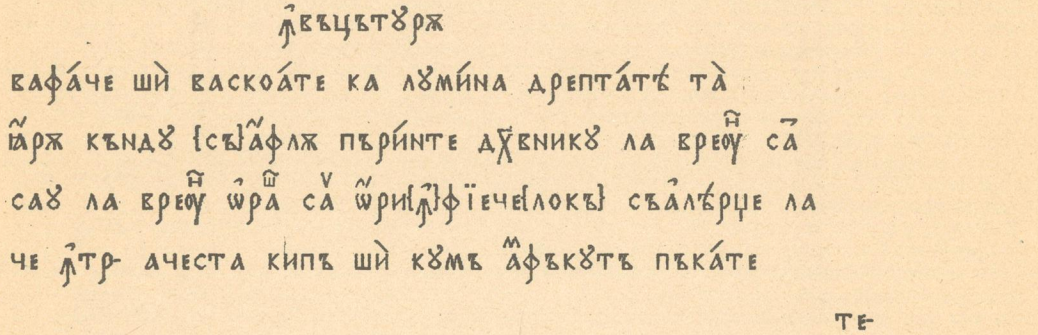


Fig. 13

Het systeem bestaat uit drie of vier afzonderlijke onderdelen:
a) Definitie en voor gebruik gereed maken van een of meer karakter-verzamelingen (boven of onder de regel geschreven tekens als accenten, onderstrepingen of cedille's vormen aparte karakter-verzamelingen). Dit betekent dat de toekomstige gebruiker zelf de afdruk-tekens ontwerpt en volgens specifikaties van medewerkers van de computer afdeling met potlood tekent op zogenaamde "schrap kaarten" (een soort ponskaart waarop de informatie niet door middel van gaatjes, maar met potloodstreepjes is aangebracht). De stapel schrapkaarten die een karakter verzameling geheel bevat wordt daarna met het computer programma KARIN in het computer-systeem gebracht en op "schijf" opgeslagen. Beperking: de karakter-verzameling mag niet meer dan 500 tekens omvatten. Een alternatief is dat de karakters op een beeldscherm worden gevormd. Op deze manier zijn de letters gevormd die u nu leest.
b) Definitie van de gewenste layout voor het afdrukken. Een programma met de naam LAYOUT stelt de gebruiker in staat in een "dialoog" via de terminal eisen vast te leggen ten aanzien van regellengte, marge, regelafstand, inspringen bij alinea's, spatie-afmeting, te gebruiken karakter-verzameling en het desgewenst rechttrekken van de rechter kantlijn door middel van het vergroten van spaties tussen woorden op de regel. Deze gegevens met betrekking tot de layout worden eveneens opgeslagen op schijf en kunnen bij meer dan een afdruk gebruikt worden.
c) Het samenstellen van de af te drukken tekst. Dit kan op een groot aantal verschillende manieren gebeuren, variërend van het inlezen van gewoon geponste kaarten tot het door een programma tekst-regels laten genereren aan de hand van zekere invoer-gegevens. Ook hier geldt weer: de gebruiker zorgt zelf voor de basis-tekst, maar kan van de computer-afdeling programmeer-hulp verwachten. Intypen van teksten kan vanzelfsprekend ook aan een terminal gebeuren. Het fraaist is het, als deze terminal "programmeerbaar" is, zodat de door een gebruiker ontworpen karakter-verzameling ook op het beeldscherm afgebeeld kan worden en dus teksten tijdens het intoetsen al in die karakter-verzameling verschijnen. Een dergelijke terminal komt januari 1981 beschikbaar.
d) Het afdrukken van de tekst. Hiervoor moet een op de wensen van de gebruiker toegesneden programma geschreven worden, dat in veel gevallen niet al te ingewikkeld hoeft te zijn (bijvoorbeeld het omkeren van de volgorde op de regel voor Hebreeuwse teksten). In sommige nu in gebruik zijnde programma's waar het gaat om het uit bepaalde invoer genereren van tekst, zijn de stappen c) en d) binnen een programma gekombineerd. Maar dan is het nog wel nodig die "bepaalde invoer" in te brengen.

D. Lexicon-systeem.

Bij onderzoek dat zich over langere tijd uitstrekt of waarbij gegevens vanuit ander onderzoek nodig zijn is een gestandaardiseerde en tegelijk flexibele manier van opslag noodzakelijk. Men denke daarbij aan: woordenboeken, indices, verzamelingen van gedigitaliseerde spraak- of muziekgedeelten, persoonsgegevens, enzovoorts. Dus in het algemeen attributen van "items", waarbij zowel attributen als items voorgesteld worden door (mogelijk meerdere) rijtje(s) van een variabel aantal tekens.
Juist dit variabele aantal is voor een groot aantal zogenaamde "data-base"-systemen een handicap, maar in een taalkundige omgeving een noodzaak. Daarom werd besloten een zogenaamde "lexicon"-systeem te ontwikkelen dat deze variabele lengte wel aankan, een efficiënt gebruik maakt van schijf-ruimte en bij het opvragen van de attributen zo snel mogelijk is. Een attribuut kan direkt achter een item worden opgenomen of als een verwijzing naar een aparte attributen-file. Gemeenschappelijke prefixen van items en attributen worden slechts een keer opgeslagen. In informatica-termen: een virtuele trie-struktuur. Bestanden die betrekking hebben op woorden uit een natuurlijke taal (zoals bv. een woordenboek met bij elk woord een variabele hoeveelheid informatie) zullen gemakkelijk een omvang van enkele tienduizenden woorden kunnen hebben. De Eclipse computer van de computer-afdeling heeft evenmin als andere computers de mogelijkheid een bestand van dergelijke omvang direkt beschikbaar in het kerngeheugen te hebben tijdens het verwerken van een programma dat van die woordenlijst gebruik maakt. Het is dus nodig de woorden op een achtergrondgeheugen te houden, dat wil zeggen op de schijf.
De tijd echter die nodig is om een gegeven van de schijf af in het programma (tijdens uitvoering) beschikbaar te krijgen is een groot veelvoud van een raadpleging van het gewone geheugen. Was het nu maar zeker, dat van elk woord precies aan het programma bekend was waar het op de schijf is opgeslagen, dan zou deze zoek-tijd wellicht nog aanvaardbaar zijn. Maar ook zo'n lijst schijf-adressen zou te omvangrijk zijn om in het geheugen te houden (nog afgezien van de onmogelijkheid om te weten welk adres bij welk woord hoort, behalve dan via een volledige lijst van woorden, die niet in het geheugen past).
In eerste instantie denkt men misschien aan een alfabetisch geordende woordenlijst op schijf, waarbij voor het opzoeken van een willekeurig woord gemiddeld de halve lijst moet worden doorlopen. Dat wil dus zeggen bij een omvang van 20.000 woorden: gemiddeld 10.000 schijf-toegangen voor het juiste woord en de erbij horende extra informatie is gelokaliseerd. Dit zou tot onaanvaardbaar lange zoek-tijden aanleiding geven en het is dus zeer zinvol slimmere oplossingen te zoeken.
Eind 1979 heeft de computer-afdeling een systeem voor het opbergen van woorden aan het werk gekregen, dat zeer bemoedigende resultaten geeft, zowel wat betreft het konstrueren van de woordenlijst als ten aanzien van het raadplegen ervan. Retrogade lexicons werden samengesteld van alle 600.000 woorden uit een corpus Nederlandse Taal ("Eindhovense Corpus"), waarin 110.000 verschillende woorden blijken voor te komen. Het opzoeken van een woord in dit lexicon (alle woorden werden weer aangeboden) kostte gemiddeld 0.05 sekonde. Aanvullende procedures zijn geschreven om programma-constructies mogelijk te maken als: doe voor alle woorden tussen woord A en woord B iets. Momenteel worden procedures geschreven om de hiërarchische struktuur van de "trie" uit te buiten: hiermee zijn bijvoorbeeld de data-bestanden van Geschiedenis te behandelen. De hierdoor ontstane boom-organisatie kan ook uitstekend gebruikt worden voor het hanteren van grote (LISP-achtige) lijst-strukturen die permanent toegankelijk moeten zijn. Vanwege de virtuele geheugen-techniek blijven de meest frekwente woorden in het (snel) geheugen. (Dit zijn ook de kortste woorden). In de literatuur is nog weinig bekend over het gedrag van deze virtuele trie-strukturen. Over de implementatie is van de hand van Joos Skolnik een congres-voordracht beschikbaar.

E. Syntactische patroon-herkenning.

In hoofdstuk 2 werd al aangeduid dat tekst-corpora voor meer doeleinden gebruikt kunnen worden dan het maken van concordanties en het doen van tellingen. In 1975 formuleerde Brandt Corstius een eenvoudige vraagtaal waarmee zinnen kunnen worden opgevraagd uit het zogenaamde Eindhovense corpus. (Zie hoofdstuk 8). Van der Steen breidde deze uit en maakte hiervoor een programma op een mini-computer van de Erasmus Universiteit, dat later is overgebracht naar de Eclipse-computer van de computer-afdeling. Twee belangrijke aspecten ervan zijn: interaktie met de gebruiker en snelheid van zoeken.
Het corpus bestaat uit een groot aantal fragmenten tekst, waarbij elk woord voorzien is van een morfologische codering van 3 cijfers dat hiërarchisch is opgebouwd. Voor het op te geven patroon geldt dat de zins-grens nooit kan worden overschreden. De kleinste eenheid die kan worden genoemd is het woord. Het systeem is "gebruikers-vriendelijk": de gebruiker behoudt ten allen tijde de controle over het systeem en kan commando's geven door het indrukken van een toets.

De patronen.

Een patroon kan worden opgebouwd uit een aantal simpele patronen door deze met de woordjes .of., .en. en .niet. te verbinden. Dit heet een "boolse" combinatie. Zijn bijvoorbeeld A, B en C simpele patronen dan betekent: "A .of. B .en. .niet. C " : zoek naar een zin die simpel patroon A bevat of simpel patroon B. In het laatste geval mag de zin niet simpel

patroon C bevatten.
Een simpel patroon bestaat in principe uit een opeenvolging van woorden, waarbij elk woord gevolgd wordt door een morfologische code. Voor een woord kan ook het zogenaamde "don't care" teken "*" gebruikt worden en voor een code het "don't care" teken "C". Op zo'n plaats accepteert het systeem elk voorkomen van een woord respectievelijk code.
Een "-" betekent: een willekeurig aantal (eventueel 0) woorden met codes. In het patroon kan een code uit minder dan drie cijfers bestaan: 23 betekent bijvoorbeeld: match elke code die met 23 begint.
Tenslotte: het teken "#" "ankert" het patroon aan het begin of aan het einde van een zin.
Voorbeelden van simpele patronen:
- 'mits 501' alle zinnen waar het woord 'mits' in voorkomt met code 501;
- 'mits C' alle zinnen waar het woord 'mits' in voorkomt;
- '# * 10 mits C - * 370' alle zinnen die beginnen met een woord waarvan de code begint met 10 dat gevolgd wordt door het woord mits en waar verderop in de zin een woord met code 370 voorkomt.
Voorbeeld van een samengesteld patroon: - 'mits C .en. .niet. (* 129 man C)' De zin bevat het woord 'mits' en niet een woord met code 129 dat gevolgd wordt door het woord man.

De commando's.

Met behulp van een aantal één-letterige commando's (beginners vergeten namelijk altijd om de return-toets in te drukken) kan het systeem gestuurd worden voor wat betreft: selectie van te doorzoeken corpora, eventueel afdrukken via een printer, al of niet afdrukken van de grammaticale codes, afdrukken van alleen een deel van een zin rondom het gevraagde patroon, onderbreken van het zoeken, het op het beeldscherm laten zien van de volgende zin die aan het patroon voldeed.
Krijgt men bijvoorbeeld zinnen die men niet bedoelde dan kan het zoeken worden gestopt en kan het patroon opnieuw worden geformuleerd. Op het beeldscherm wordt voortdurend aangegeven hoeveel zinnen er zijn gevonden die nog niet door de gebruiker op het beeldscherm zijn bekeken (" zoveel wachtenden voor u").

Voorbeelden.

Het programma kan werken op corpora waarvan de woorden zijn voorzien van morfologische coderingen. De snelheid van zoeken is gemiddeld 2400 woorden / seconde.
Corpora zonder coderingen (dus elke willekeurige tekst die in zinnen is onder te verdelen) zijn op dezelfde manier te gebruiken (de vakgroep Frans bijvoorbeeld zocht op deze manier in de "Fabliaux"). Voor corpora met uitgebreidere coderingen wordt momenteel een algemener systeem ontworpen waarin ook de mogelijkheden voor de patronen aanzienlijk wordt vergroot. (Verdere suggesties daarvoor zijn van harte welkom).
In feite groeien dit programma en het hiervoor beschreven programma LALR naar elkaar toe. Van de hand van Van der Steen is hierover een congres-voordracht beschikbaar.
Om een goed beeld te krijgen van de bruikbaarheid van het programma kan men het beste achter een terminal plaatsnemen. Anderzijds kan de computer-afdeling een uitdraai verzorgen wanneer u reeds zeker bent van de patronen waarnaar u wilt zoeken.
We geven hieronder twee voorbeelden: een uit het Eindhovense Corpus (fig. 14) en een uit het Brown Corpus (fig. 15). Het eerste patroon is afkomstig van Brandt Corstius (vakgroep ATW). Het beschrijft Nederlandse zinnen die een tussenwerpsel (code 800) bevatten dat echter niet het woordje "ja" en ook niet het woordje "nee" mag zijn. Het tweede patroon is afkomstig van Meijs (vakgroep Engels). Om zijn woorden te gebruiken: "The pattern singles out cleft sentences in which the element in focus functions as object of the relative clause. Meaning of the patterns-elements:

IT PPS	"IT" as personal pronoun singular
* BE	any form of the verb "be"
-	any intervening material
* N	any noun
* V	any verb
* WPO	any WH-pronoun in object function."

* 800 .EN. .NIET. JA 800 .EN. .NIET. NEE 800

---------- CORPUS 1 ----------

116 Welnu 800 , zij 300 hebben 254 hun 330 inspraak 000 , medezeggenschap 000 en 700 democratisering 000 .
141 Pardon 800 , ik 300 ben 271 de 370 draad 000 van 600 deze 370 zin 000 een 450 beetje 000 kwijt 100 .
542 Ik 300 ben 271 Brychczy 010 , jazeker 800 , " klonk 245 het 440 in 600 brokkelig 100 Duits 120 , " majoor 000 Brychczy 010 , jazeker 800 . "
546 Ach 800 , op 600 een 450 veld 000 als 720 dit 360 was 275 een 450 verdedigende 203 partij 000 natuurlijk 150 in 600 het 370 voordeel 000 .
618 Welnu 800 , dan 510 neem 251 ik 300 dat 370 geld 000 ergens 540 vandaan 610 .
643 Daaraan 540 werd 275 met 600 het 370 lied 000 : " O 800 , Dick van Dijk 010 , wat 410 ben 272 je 300 diep 150 gezonken 206 " , trouwens 500 duidelijk 150 uiting 000 gegeven 216 .
1204 Niet 500 minder 454 menselijk 100 , integendeel 800 .
1666 Sommige 453 lieden 001 letten 244 niet 500 op 600 de 370 eetgewoonten 001 van 600 een 450 aap 000 , zoals 710 ie 300 zijn 330 pinda's 001 pelt 253 , het 370 papier 000 van 600 het 370 snoepje 000 doet 253 , die 360 zeggen 254 heel 500 hard 150 : " H!e 800 , kijk 247 , apen 001 ! 998 Die 360 hebben 254 rooie 103 konten 001 of 700 ze 300 stinken 244 . "
1675 Welaan 800 dan 510 .
2447 Waar 150 gebeurd 206 , nietwaar 800 , mamma 000 ? "
2451 Och 800 , ik 300 kan 271 zoveel 440 vertellen 210 , maar 700 ik 300 moet 271 straks 500 gaan 230 rusten 210 .
2455 Maar 700 het 440 moet 243 nu 510 eenmaal 500 he 800 . . 998 ?
2473 De 370 president 000 van 600 de 370 Rechtbank 000 in 600 Groningen 010 , mr. G. Overdiep 010 , heeft 273 de 370 inbrekers 001 , die 420 hij 300 gistermorgen 500 voor 600 zich 340 kreeg 255 allemaal 440 ingeprent 216 : " Heus 800 , ik 300 zou 275 er 540 maar 500 mee 610 ophouden 200 .
3153 Maar 700 zelfs 500 daar 540 hoeft 273 nog 500 geen 450 man 000 hoera 800 tegen 610 te 650 roepen 210 en 700 vanmiddag 500 beet 255 onze 333 kamerjuf 000 in 600 het 370 hotel 000 , een 450 welgedane 103 dame 000 van 600 middelbare 103 leeftijd 000 , haar 303 toe 620 : " Mens 000 je 440 ziet 243 er 510 tien 470 jaar 001 ouder 104 uit 620 " .
3330 Dat 360 kan 273 gebeuren 200 , nietwaar 800 ?
3544 Hup 800 , de 370 tijdmachine 000 in 630 (het 440 is 273 300 460 na 600 Christus 010) en 700 dan 510 , na 600 bezoekjes 001 aan 600 Albinoni 010 , Bach 010 en 700 Tsjaikowsky 010 , per 600 raket 000 verder 154 de 370 Kosmos 000 en 700 de 370 Toekomst 000 in 630 , op 600 de 370 tonen 001 van 600 " mindblowende 103 " muziek 000 .

Fig. 14

(IT PPS * BE - * N * WPO) .AND. .NOT. (IT PPS * BE - * V * V * N * WPO)

---------- CORPUS C ----------

890 - 10 he has married me with a ring of bright water , begins the Kathleen Raine poem from which Maxwell takes his title , and it is this mystic bond between the human and natural world that the author conveys .

---------- CORPUS G ----------

2100 - 21 but it is the need to undertake these testaments that I would submit here as symptom of the common man's malaise .
2560 - 26 if many of the characters in contemporary novels appear to be the bloodless relations of characters in a case history it is because the novelist is often forgetful today that those things that we call character manifest themselves in surface behavior , that the ego is still the executive agency of personality , and that all we know of personality must be discerned through the ego .
6441 - 68 it is not possible to reconstruct fully the arrangements whereby these honors lists were then made up or even how the names that they contained assumed the order in which we find them .

---------- CORPUS J ----------

3675 - 41 it is this conclusion that we challenge ;
4598 - 52 and if we can see that what we meant to say remains the same , while the feeling varies from intensity to near zero , it is not the feeling that we primarily meant to express .

---------- CORPUS K ----------

1244 - 08 for it was Rameau's type of music that he had been trying to write , and that he couldn't write .

Fig. 15 (coderingen niet afgedrukt)

Hierna volgt als illustratie van een gecombineerd gebruik van de programma's uit B, D en een beschrijving van een toekomstig, omvangrijk projekt.

De computer en het Engels Seminarium.

Bij het Engels Seminarium bestaan plannen voor een drietal projekten binnen het aandachtsgebied Computer Corpus Studies. Hierbij wordt enerzijds voortgebouwd op 4 jaar ervaring binnen de interuniversitaire werkgroep CCPP (Computer Corpus Pilot Project), anderzijds wordt gebruik gemaakt van materiaal dat de laatste jaren is aangeworven:
- een drietal corpora: Brown, LOB, en Lund
- en een woordenboek (de ALD) op computer-tape.

Het centrale projekt, An Expanding Lexical Data Base, beoogt het ontwikkelen van een flexibel systeem voor de opslag van lexicale informatie, waarbij de ALD, en de lexicons die uit de verschillende corpora kunnen worden geabstraheerd als basis dienen. De grammatical codering zal grotendeels het CCPP-systeem volgen, aangevuld met sub-categorisatie-gegeve zoals in de ALD (The Advanced Learner's Dictionary, OUP).

Het opbouwen van een uitdijend lexicon heeft mede ten doel het ontwikkelen van een zoveel mogelijk geautomatiseerd systeem voor het coderen en analyseren van Engelse teksten. Het werk hieraan wordt ondergebracht in het projekt Analytic Procedures. Basis-hypothese is dat bij een rijke lexical data base uiteindelijk kan worden volstaan met het manueel codere van alleen werkwoordelijke elementen, en het markeren van konstituenten die meer dan een woord bevatten. Hierbij wordt uitgegaan van een zo "plat" mogelijke zins-struktuur (zoals de CCPP-benadering).

In het derde projekt, Corpus Grammar, vinden deel-onderzoeken naar verschillende grammaticale (alsook lexicale) onderwerpen plaats, waarbij gebruik gemaakt wordt van de corpora die al beschikbaar zijn. Het gaat hierbij vooral om onderwerpen die in moderne theoretische benaderingen onvoldoende aan bod komen en om de toetsing van theoretische benaderingen aan empirische gegevens. Een tweetal scripties, een over "clefts and pseudo-clefts" en een over "appositions" is onlangs afgerond, en er is een proefschrift-studie over "collocatie-restricties" aan de gang. Verder wordt gewerkt aan he opbouwen van een "pool" van gericht descriptief- / illustratief materiaal op basis van computer-corpus gegevens voor didaktische doeleinden.

Meest betrokken staflid bij de verschillende hier genoemde aktiviteiten is Meijs. Aan het CCPP-projekt heeft ook Bienen meegewerkt. Verder zijn Verdonk en Van Brederode bij verschillende onderdelen betrokken. Vier student-assistenten hebben aktief aan het CCPP-werk meegedaan, te weten Tax, Peeters, Hagemeijer en Van Gelderen; de laatste heeft in een uitgebreide scriptie, "Head-Hunting: An Approach to Phrases in a Computer Corpus", enige verbeteringen op het CCPP-codeersysteem voorgesteld en uitgewerkt, alsook een eerste aanzet gegeven tot het automatisch "labellen" van de veschillende typen "phrasal constituents". Het is de bedoeling dat voor de uitvoering van het "lexical data base" projekt een onderzoeks-medewerker wordt aangezocht. Voorts zal aan de verschillende projekten worden meegewerkt door doktoraal-studenten Modern Engels in het kader van de doktoraal-werkgroep "Corpus Studies".

Nadere gegevens kan men vinden in:
Werkgroep CCPP: Manual for Coders, Nijmegen 1978
W.J. Meijs: "Cracking the Code", English Institute Journal 5, december 1980
H. v. Gelderen: Head-hunting: An Approach to Phrases in a Computer Corpus, scriptie Engels Seminarium UvA, 1980.

W.J. Meijs

Hoofdstuk 6. Onderwijs op de computer-afdeling.

De computer-afdeling houdt zich naast dienstverlening t.b.v. onderzoek en naast eigen onderzoek ook bezig met onderwijs. Dit onderwijs is in de eerste plaats bedoeld voor studenten en medewerkers van de faculteit die zich de vaardigheden willen verschaffen die nodig zijn om de computer bij hun onderzoeks-projekt(en) te gebruiken. Daarbij is niet uitsluitend gedacht aan de eigen machine van de computer-afdeling. De strekking van de kursussen is ruimer, zodat de verworven kennis ook op andere installaties kan worden toegepast.

De verschillende kursussen die gegeven worden zijn:
- Algorithmen en Programmeren in PASCAL (in samenwerking met ATW)
- DG/L
- Automatische Syntaxherkenning.

De kursussen zullen hieronder kort besproken worden.

Algorithmen en Programmeren in PASCAL

Deze kursus heeft tot doel de kursisten drie dingen bij te brengen:
a. Het algorithmiseren van taken.
b. Het schrijven van programma's in de programmeertaal PASCAL.
c. Het lezen en analyseren van door anderen geschreven programma's in PASCAL.
De nadruk wordt gelegd op het onderdeel algorithmiseren. Dit kan worden omschreven als: Het opstellen van een "recept" in menselijke taal voor het laten uitvoeren van een zekere taak (die door een machine zou moeten kunnen worden volbracht). Het "recept" moet bestaan uit opdrachten die elk een deel-taak omschrijven, zodanig dat de totale taak wordt uitgevoerd als de deel-taken in de voorgeschreven volgorde achtereenvolgens worden volvoerd. Daarbij worden beperkingen opgelegd aan de formuleringen. Opdrachten kunnen zijn van één van drie typen: direkte opdrachten, konditionele opdrachten en herhalings-opdrachten. Direkte opdrachten mogen naar wens geformuleerd worden, mits ondubbelzinnig een (deel-) taak omschrijvend.
Konditionele opdrachten moeten zijn van de vorm:
"Indien" zekere konditie "dan" een of meer opdrachten "anders" een of meer opdrachten. Het gedeelte vanaf (en inclusief) "anders" mag worden weggelaten.
Herhalings-opdrachten hebben de vorm:
"Zolang" zekere konditie "doe dan" een of meer opdrachten. De enkele deel-taken die, mits achtereenvolgens uitgevoerd, de totale taak volbrengen worden met deze taalelementen beschreven, en vormen aldus het "grofste algorithme".
De direkte opdrachten mogen zo ruim geformuleerd zijn dat ze zelf weer kunnen worden opgedeeld in deel-taken, en dus weer beschreven kunnen worden als recepten op zichzelf met de zelfde taalelementen. Een herhaald toepassen van dit proces van "stapsgewijs verfijnen" levert tenslotte een gedetailleerd algorithme op dat vrijwel woord voor woord te vertalen is in een van de hogere programmeertalen. Immers de equivalenten van de algorithmische taalelementen voor gestruktureerde opdrachten (konditioneel- en herhalings-) komen in elke goede hogere programmeertaal voor.
Dit algorithmiseren is tot het belangrijkste onderdeel geworden van de kursus, omdat het de grondslag van elk gebruik van een machtig maar dom instrument als een computer is.
De taal PASCAL wordt onderwezen, omdat dit een goede hogere programmeertaal is en tevens omdat het een "didaktische" taal genoemd kan worden. De meeste andere programmeertalen stellen programmeurs naast de taalelementen ook de mogelijkheid ter beschikking, oneigenlijk gebruik van die elementen te maken. Dat leidt tot voor buitenstaanders soms onbegrijpelijke programma's. PASCAL beperkt zulk oneigenlijk gebruik, zodat de kursisten getraind worden in korrekt gebruik van de mogelijkheden die een programmeertaal biedt. Als ze later een "slordiger" taal tot hun beschikking krijgen zal de verworven discipline ze, naar wij hopen, op het smalle pad der leesbare programmatuur houden.
In de praktijk van het omgaan met computers komt het nogal eens voor dat programma's die door anderen zijn gemaakt enigszins moeten worden aangepast voor een specifieke toepassing. Daarvoor is het nodig geoefend te zijn in het lezen en analyseren van door anderen geschreven programma's. Als afronding van de kursus hebben we dit onderdeel dan ook toegevoegd aan de behandelde onderwerpen.
De kursus wordt in principe tweemaal per jaar gegeven indien er voldoende belangstelling voor bestaat. De vorm is: een blokkursus van een week van hele dagen aan het begin van een semester gevolgd door een wekelijkse bijeenkomst van 2 uur gedurende het semester.

DG/L

DG/L (spreek uit: dee-gee-el) is een programmeertaal die de fabrikant (Data General) van de machine van de computer-afdeling heeft laten ontwerpen voor gebruik op Data General computers. Deze "Data General Language" vertoont grote overeenkomsten met bijvoorbeeld Algol 60, maar is verrijkt met veel elementen van PL/I (pee-el-een). Daardoor wijkt de taal niet al te sterk af van deze meer algemene programmeertalen die niet aan een merk zijn gebonden. (Overigens wijken ook de door fabrikanten geleverde versies van algemene programmeertalen dikwijls af van de specifikaties zoals die voor de officiële versies gelden).

Een student met enige ervaring in het schrijven van programma's (bijvoorbeeld na het volgen van de kursus Algorithmen en Programmeren in PASCAL) kan zich vrij eenvoudig het idioom van DG/L eigen maken. Daardoor krijgt hij een aantal programmatische hulpmiddelen tot zijn beschikking die hij in een andere taal zelf zou hebben moeten programmeren. Een groot deel van die hulpmiddelen is onderdeel van de taal DG/L, maar een ander deel is ontwikkeld binnen de computer-afdeling.
Voorbeelden van DG/L elementen die niet in bijvoorbeeld PASCAL voorkomen:
-de data-struktuur STRING (een rij tekens van willekeurige en variabele lengte)
-een aantal standaard functies en procedures om deze STRINGs te behandelen.
Juist deze elementen maken DG/L zo bruikbaar in onze faculteit, waar een groot deel van het werk bestaat uit het analyseren van teksten.
Daarnaast wordt DG/L gebruikt voor het meer specialistische systeem-programmeren: elke funktie van het operating-systeem kan worden benut (zoals multi-tasking en snelle I/O), terwijl het werken met bitstrings (bijvoorbeeld voor het plotten van karakters) door de direkte vertaling naar de daarvoor beschikbare machine-instrukties heel efficiënt werkt. De kursus wordt op ongeregelde tijden gegeven als er voldoende vraag naar is. Hij duurt ongeveer een week van halve dagen. Programmeer-ervaring wordt verondersteld.

Automatische Syntaxherkenning

Zowel in de informatica als in de formele taalkunde spelen grammatica's een belangrijke rol om de syntaxis van een taal te kunnen beschrijven. Het blijkt lang niet altijd gemakkelijk om programma's te schrijven die aan de hand van deze grammatica's een zin kunnen ontleden. Voor de meest bekende categorie, de kontekst-vrije grammatica's, zijn genoeg methoden voorhanden waarmee in een redelijke tijd alle boom-strukturen van een zin geconstrueerd kunnen worden. Natuurlijke taal is echter ten dele kontekst-vrij te beschrijven. In de kursus wordt getoond hoe men in de grammatica patronen kan opnemen die het herkennend vermogen opvoeren zonder dat de ontleedbaarheid (en de snelheid van ontleden) in het gedrang komt. Men wordt vertrouwd gemaakt met de werking van de parser-generator van de computer-afdeling.
Aansluitend worden recente ontledings-methoden besproken voor sub-klassen van kontekst-gevoelige talen.
Voor het volgen van de kursus is programmeer-ervaring in Algol of Pascal gewenst, evenals enige kennis van formele talen en automaten.
Om studiepunten te behalen dient men een aansluitend praktikum te volgen.

Hoofdstuk 7. Hoe kom ik aan de slag ?

Het antwoord is eenvoudig: u maakt een praatje met een van de medewerkers van de computer-afdeling: Herengracht 338, onder het Fonetisch Instituut (dus het trapje af; de deur is open), telefoon (020) 5252181. Er is altijd iemand aanwezig van 9-21 uur.
(N.B.: telefoneert men van buiten een doorkiesnummer dan hoort men in plaats van een bezettoon de kiestoon).
Al naar gelang uw noden en wensen wordt u als volgt ingedeeld:

- u kunt volstaan met het gebruik van de aanwezige apparatuur en programmatuur. Wij proberen u daar dan vertrouwd mee te maken. Data-invoer moet echter door u zelf verzorgd worden. Hiervoor kunt u een draagbare terminal met cassettes in bruikleen krijgen, of u kunt verwezen worden naar een ponsbureau.
Gebruikt u afwijkende lettertypen dan kunt u ook invoeren aan een terminal met programmeerbare karakter-verzameling.

- er moeten kleine veranderingen gemaakt worden in een bestaand programma. U bespreekt deze met de medewerker die het programma beheert.

- u kunt direkt geholpen worden met apparatuur en/of programmatuur en/of specifieke assistentie elders, bv. bij het Technisch Centrum van de FSW of bij het Mathematisch Centrum (wat vaak voorkomt bij statistische problemen). U wordt dan zo goed mogelijk doorverwezen.

- voor u moeten speciale programma's gemaakt worden die niet elders voorhanden zijn en die gedurende langere tijd beslag zullen leggen op een medewerker. U zult dan verzocht worden om een globale schets van uw onderzoeks-plannen in te dienen bij de facultaire computer-commissie.
U wordt echter het snelst geholpen als u zichzelf helpt: d.m.v. de programmeer-kursussen en de aansluitende praktika kunt u zelfstandig werken. Incidentele programmeer-adviezen worden dan graag gegeven.

- u hebt plannen voor de automatisering van uw studenten-administratie, uw boekhouding of uw bibliotheek. In dat geval zult u gewezen worden op de centrale faciliteiten van de universiteit en de afspraken die daarover bestaan. Zowel de dienst Financiële Administratie (FAS) als de Universiteits Bibliotheek kennen een automatiserings-afdeling. Coördinator van alle automatiserings-aktiviteiten bij onze universiteit is de heer Van Eerde, tel. 5252738.

Hoofdstuk 8. Bibliotheek en tekst-bestanden.

De computer-afdeling beschikt over een eigen bibliotheek die is ondergebracht in de bibliotheek van het Fonetisch Instituut.

Lopende tijdschrift-abonnementen zijn:
- ALLC Bulletin (Association for Literary and Linguistic Computing), vanaf 1973
- BYTE the small systems journal, vanaf 1980
- Computer Music Journal, vanaf 1981
- Computers and the Humanities, vanaf 1979
- IEEE Transactions on Computers , vanaf 1979
- IEEE Transactions on Pattern Analysis and Machine Intelligence , vanaf 1979
- ITL Revieuw of Applied Linguistics, vanaf 1981
- Journal of the Association for Computing Machinery, vanaf 1974
- ICAME Newsletter

Aangevraagd, maar nog niet ontvangen:
- American Journal of Computational Linguistics

Er zijn momenteel plus minus 150 boeken aanwezig.
Verdere literatuur kan geraadpleegd worden op
- het NOVI (Nederlands Opleidings Instituut voor Informatica), Stadhouderskade 6 (administratief georienteerd)
- het Mathematisch Centrum, Kruislaan 413 (wiskundig georienteerd).

Momenteel zijn op de computer-afdeling twee morfologisch gecodeerde corpora aanwezig:

- het Eindhovense Corpus (720.000 woorden), onderverdeeld in 5 sub-corpora geschreven taal:
1 dagbladen
2 opiniebladen
3 gezinsbladen
4 populair-wetenschappelijke boeken
5 romans en novellen
en 2 sub-corpora gesproken taal:
6 informeel
7 formeel.

In de laatste twee kan bovendien een selectie gemaakt worden naar leeftijd, opleiding en geslacht.
De sub-corpora 1 tot en met 5 zijn beschreven in "Woordfrequenties in geschreven en gesproken Nederlands", red. P.C. Uit den Boogaart; Oosthoek, Scheltema & Holkema, Utrecht 1975.
De sub-corpora 6 en 7 in "Spreektaal. Woordfrequenties in gesproken Nederlands", red. E.D. de Jong; Bohn, Scheltema & Holkema, Utrecht 1979.

- het Brown Corpus (1.000.000 woorden), naar genre onderverdeeld in 16 sub-corpora.

Daarnaast is aanwezig het London Lund Corpus, ("brits" engels), bedoeld als tegenhanger van het Brown Corpus en even groot met dezelfde indeling. Met de morfologische codering is nu een aanvang gemaakt.
Voor het gebruik van de twee engelse corpora dient een gebruikers-verklaring getekend te worden. Daarnaast zijn een aantal tekst-bestanden in gebruik bij diverse onderzoekers binnen de faculteit.
Bij de computer-afdeling liggen lijsten ter inzage van corpora en tekst-bestanden die elders beschikbaar zijn (sommige gratis).

Hoofdstuk 9. Apparatuur en programmeertalen.

De computer-afdeling beschikt over een eigen mini-computer die is verbonden met de CDC-computer van SARA (het rekencentrum van de UvA, VU en het Mathematisch Centrum). Vooral het interaktief zoeken naar patronen in teksten is zo computer-intensief dat dit een eigen computer rechtvaardigt.
(Er zijn nog andere voordelen, zoals de beschikbaarheid, de specifieke machine-instrukties voor string-manipulatie, de verhouding kosten / baten en het in eigen hand hebben van de bureaucratie).

Per 1 februari 1981 is de configuratie van de mini:
- 256 KB geheugen
- 200 MB disc (vooral t.b.v. tekst-bestanden)
- 10 MB disc (vooral t.b.v. signaalanalyse)
- 800 BPI mag. tape unit
- 200 c/min card-reader
- Calcomp plotter
- Versatec 1200A printer / plotter
- Data Products line-printer (450 lpm)
- asynchrone multiplexer
- 4 displays (2 Superbee's en 2 Minibee's)
- 1 Tektronix 4025 display t.b.v. de invoer in afwijkende lettertypen en t.b.v. speech-editing (fonetiek)
- 1 draagbare terminal met cassettes, ingebouwde editor en modem (Miniterm)
- synchrone 4800 baud verbinding met SARA t.b.v. 200-UT protocol.

Systeem:
- Time-sharing systeem (TSS, van Polimorphic Inc.) dat draait in een partitie van een real-time operating system (RDOS, van Data General).

Programmeertalen:
- DG/L. Vooral in gebruik voor programma's die alles uit de machine halen en t.b.v. string manipulaties. (Zie beschrijving in hoofdstuk 6).
- PASCAL. Vooral t.b.v. onderwijs (elk semester wordt weer bezien of het praktikum op SARA of op de Mini zal plaatsvinden)
- FORTRAN. Vooral t.b.v. fonetiek en musicologie
- BASIC. Vooral t.b.v. het "even een programma'tje maken".
- LISP. Incidenteel ten behoeve van A.I.-programma's.

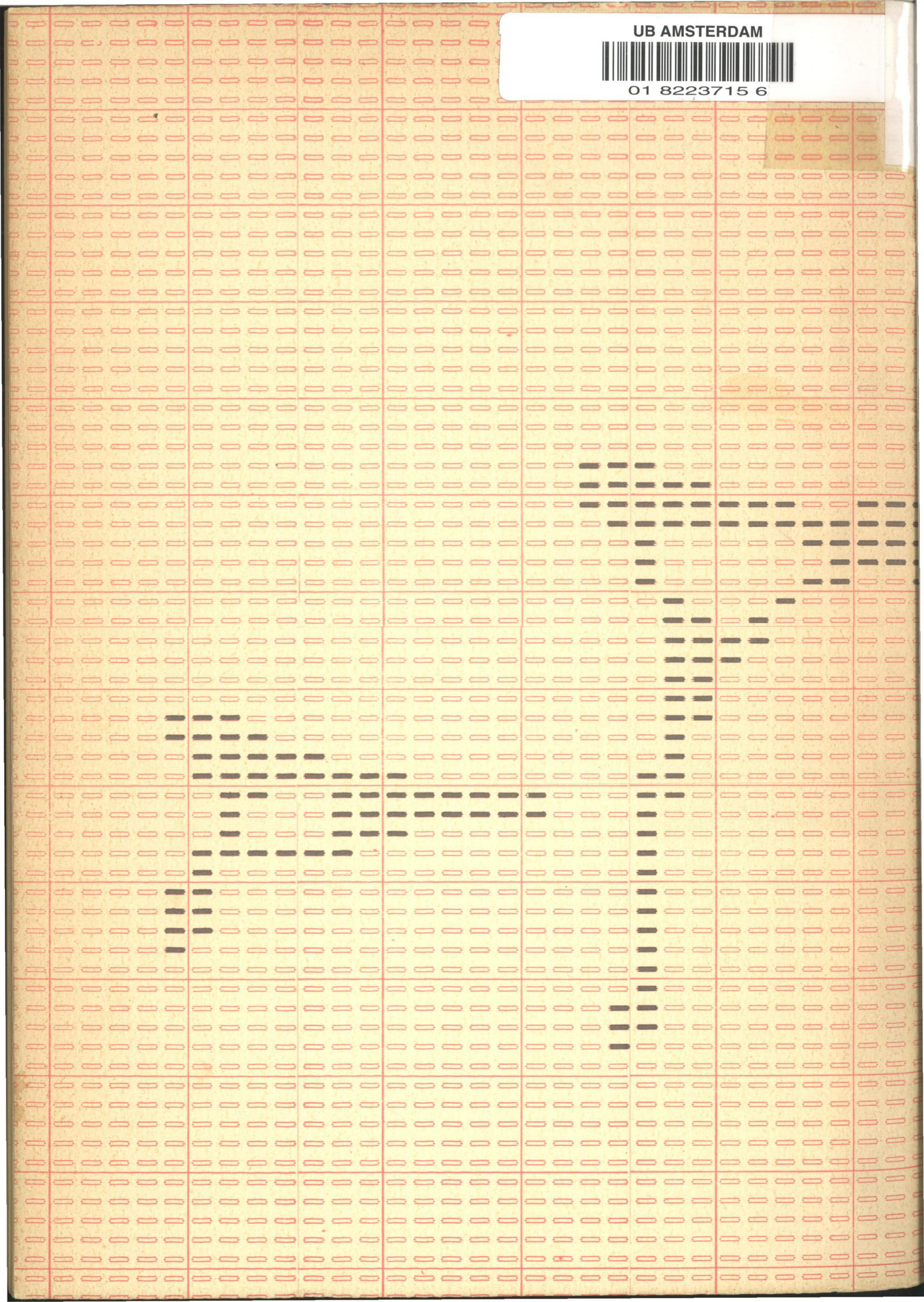
Daarnaast de noodzakelijke utilities, waaronder genoemd kunnen worden een krachtige editor en een tekst-opmaak programma.

Op SARA wordt vooral gebruik gemaakt van SPSS (statistische verwerking). Verder incidenteel: RIQS, Algol 68, Snobol. De gebruiks-mogelijkheden van SIR worden momenteel onderzocht. Het omzetten van magneetbanden naar andere dichtheden en het afdrukken van zeer grote tekst-bestanden gebeurt eveneens op SARA.

Index.

aanvraag-procedure 4
achtergrondgeheugen 37
ad-konversie 23
adviescommissie 3
ai-programmas 6
alfabetische 13 37
alfabetiseren 20
alfabetten 36
algol 32 42 43 46
allc 45
allc-bulletin 7
ambiguiteiten 35
anaforische 32
analoog 23
analoog-digitaal-con 30
arabische 30 31
archeologie 7 9
archivering 9
artikulatoren 23
assimilatieregels 14
assistentie 12 44
astur 3
atw 33 38 42
automaten 8 35 36 43
backtracking 35
basic 46
batch 5
beheer 6 34
bestands-grammaticas 12 34
bestuur 3
bitstrings 43
blokkursus 42
boedelinventarissen 12
boom-organisatie 37
boom-strukturen 35 43
brown 41 45
byte 45
calcomp 46
card-reader 46
cassettes 44 46
catalogisering 9
ccpp-projekt 41
cdc 32 46
cf-grammaticas 35
chinees 7
cic 3
cluster-analyse 9
cocofac 3
coling 7
computer-commissie 3 4 44
computer-configurati 30
computer-gebruik 2 3 6
computer-geheugen 5 7
computer-taalkunde 8
configuraties 3 46 30
congres 7 8
congres-voordracht 37 38
corpus 41
criva 3
cyber 32
cybernetika 24
cyrillische 36
dagbladen 45
data-base 5 6 9 34 37
data-beheer 12
data-bestanden 9 35 37
data-invoer 12 44
data-strukturen 34 43
declinaties 20
dgl 42 43 46
dialectologie 11
didaktische 41 42
dienstverlening 3 42
digitaal 23
disambiguering 8
displays 46
drukken 36 38
drukkerijen 7
duits 20 39
eclipse-computer 35 37
editor 46
eerde, van 2 3 44
eindhovense corpus 35 37 38 45
fabliaux 13 14 38
faculteitsraad 3
fas 44
formanten 23 24 25
fortran 46
fourier 23 24 30
frekwenties 14 24 31
fsw 44
gebruikers-vriendeli 37
gegevens-bestanden 5 35
gegevens-strukturen 32
geheugen-ruimte 5
geheugen-techniek 37
gelofte-teksten 16
geluidsgolven 23 30
genealogie 8
germaanse 11
geschiedwetenschap 12
gezinsbladen 45
gothische 7
grafisch 9
grammaticale 8 16 38 41
handschriften 13 14
hebreeuwse 36
herkennings-programm 34
hettitisch 3 16 35
ibm 3
ifa 23 24
ils-programma 24
index 1 7 8 22
index-woorden 9
informatica 4 37
intelligentie 11
invoer-validatie 34 35
iti 45
iwi 3
kerngeheugen 37
klankvoorstelling 30
klasse-aanduiding 14
kleitablet 16
kontekst-gevoelige 43
kontekst-vrije 34 35 43
kunst-talen 8
latijnse 11
layout 36
leestekens 36
lemma 7 8 16 20 22
letterbeeld 7
letterkunde 11
letters 7 9 36
lettertypen 7 34 36 44 46
lexicografie 7
lexicologie 7
lexicometrie 7 8
lexicon-systeem 35 37
line-printer 7 46
linguistiek 7 32 45
lisp 46
lisp-achtige 37
lpc 23 24
lpc-spektra 25
lpc-spektrum 24
lund 41 45
magneetbanden 7 46
manuscript 13 14 16
micro-computers 5
middeleeuwse 13 14 20

mini-computer 4 5 37 46
minnesanger 20
modem 46
moeder-kind 9
mond-keelholte 10 23 24
morfologie 32
multi-tasking 43
multiplexer 46
musicologen 30 31
musicologie 7 9 46
muziek 9 30 31 39
muziekschrift 30 31
muzieksignaal 30
muziekstukken 30 34
muziekwetenschap 11 30
natuurkunde 30 31
neerlandistiek 3
nijmegen 3 11 41
notenschrift 30
novellen 45
novi 45
ontleder 8 35
operating-systeem 43
opiniebladen 45
orthografie 20
oud-franse 13 14
oudgriekse 11
parser-generator 35 36 43
pascal 42 43 46
pathologie 9
patroon-grammaticas 35
patroon-herkenning 9 37
plato-systeem 9
pli 42
plotten 9 43
plotter 7 46
ponsband 7
ponsmachine 35
praktikum 43 44 46
predikaatschemas 32
prediktie 23 24
printer 38 46
printer-plotters 7
programma-pakketten 35
programmeer-kursusse 44
programmeertalen 5 6 34 42 46
promovendi 41
pseudo-clefts 41
query 14
randapparatuur 3 9
rapportage 2
rapporten 35
rdos 46
real-time 46
rechts-recursieve 35
reguliere 35
reimwortverzeichnis 20
rekencentrum 3 46
retrogade 37
rijmwoorden 20
riqs 46
roemeense 11 36
romans 45
romeinse 7
sample 23 24 32
schijf 4 34 36 37 46
schriftbeelden 7
scribe 2
scripties 41
semantiek 7 8
seminarium 41
sir 34 46
slavische 11
snobol 46
software 6
software-huis 3
sorteerwerk 11
sortering 16
spaanse 11
speech 24
speech-editing 46
spijkerschrift 16
spraakanalyse 23 24
spraaksignaal 10 23 24
spreektaal 45
spss 9 34 35 46
stochastisch 10
sub-corpora 45
syllabair 16
symbolen 35 36
syntactische 8 16 32 37 42 43
syntaxherkenning 42 43
systeembeheerder 3
tape 30 46
taqsim 30 31
tekst-bestanden 45 46
tekst-corpora 8 16 34 37
tekst-grammaticas 9
tekst-opmaak 2 34 46
tekst-regels 36
tekst-stambomen 8
tekst-uitgave 20
tekstinterpretatie 11
tektronix 46
tijdschrift 7 45
tijdstruktuur 30
timbre-bepalend 30
time-sharing 4 5 46
tonen 30 31 39
tong 23
toonduur 31
toonhoogten 30
trie 37
trillingen 24 30
tsjaikowsky 39
tss 46
turing-machine 5
utilities 46
valid 35
valin 35
versatec 46
vertalen 6 7 30 42
vertalers 9
volksliederen 30
vraag-programma 8
vraagtaal 5 8 37
vraagterm 32
woordenboek 34 37 41
woordfrequenties 45
woordsoort 20 22
woordtekens 16
woordvolgorde 8
woordvormen 20
zet-instrukties 7
zins-grens 37
zins-struktuur 41
zoek-programma 16
zoek-tijden 37

JNST 125. Cat. 12.1081

UB AMSTERDAM
01 8223715 6